Office de la propriété intellectuelle du Canada

Un organisme d'Industrie Canada

Canadian Intellectual Property Office

An Agency of Industry Canada

Bureau canadien des brevets

Certification

La présente atteste que les documents ci-joints, dont la liste figure ci-dessous, sont des copies authentiques des documents déposés au Bureau des brevets.

Canadian Patent Office

Certification

This is to certify that the documents attached hereto and identified below are true copies of the documents on file in the Patent Office.

Specification and Drawing, as originally filed, with Application for Patent Serial No: **2,397,389,** on August 9, 2002, by **DJEMEL ZIOU, PIERRE POULIN, MARIE-FLAVIE AUCLAIR-FORTIER AND MADJID ALLILI,** for "Image Model Based on N-Pixels and Defined in Algebraic Topology, and Applications Thereof".

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Sylvie Grégoire
Agent certificateur/Certifying Officer

September 11, 2003
Date

Canada

(CIPO 68)
04-09-02

OPIC CIPO

ABSTRACT OF THE DISCLOSURE

A data structure in which image is specified by its support, its quantities that are linked to the support and allowable generic operations is described herein. The algebraic structures of the support are defined using algebraic topology concepts such as chain and boundary. The quantities may be a scalar, vector, tensor or any other type. They are specified by the cochain. The generic operation corresponds to the coboundary operator. The image model has several advantages: it allows the derivation of efficient algorithms that Operate in any dimension and the unification of mathematics and physics to solve classical problems in image processing and computer vision. It can be used for binary images as well as for several image acquisition systems.

TITLE OF THE INVENTION

Image Model based on n-pixels and defined in algebraic topology, and applications thereof

FIELD OF THE INVENTION

[0001] The present invention relates to an image model based on n-pixels. More specifically, the present invention is concerned with a) an image model based on n-pixels; b) an application the image model for the resolution of diffusion and optical flow; and c) an application of the image model for the deformation of curves.

BACKGROUND OF THE INVENTION

[0002] People have a notion of what an image is. For instance, for psychologists the image is linked to the shape of objects, their depth, the relationship of these shapes and their perceptual organization.

[0003] Artists are focused on how features such as shape, color, and perspective are organized to represent a scene that may originate in their imagination.

[0004] Physicists are concerned with the physical phenomena produced by a given scene and how they are represented in the image.

[0005] Neurophysiologists regard images through visual phenomena in humans and animals, such as contrast sensitivity, Mach bands, contrast, constancy, and depth perception.

[0006] While a precise definition of the image is elusive, it is clear that for certain people an image is the visualization of physiological, perceptual, or physical phenomena and for others it is related to semantic content. Whatever the term image means, the intention is to establish a foundation upon which images of all forms and contents can be discussed with minimal confusion.

[0007] In image processing and computer vision, the foundation is related to the understanding of the image formation process. Light generated by a source is modified by the objects of the scene. The modified light is captured by a system acquisition device, transformed into an appropriate form and displayed on a physical support. The content of the resulting image and, consequently, its further use, both depend on the properties of the light (structured or not, spectral properties such as the range of wavelengths, the number of ranges and the intensity), the characteristics of the acquisition device, the transformation (analog/digital, pre-processing), the display format (temporal or spatial organization of image elements, film, vector, raster). From the automatic processing point of view, the image is enhanced to improve its perceptual quality or to make some of its features explicit. Usually, its content is analyzed via a successive reduction process to construct a more descriptive representation in terms of relevant features, which can be used more effectively by a decision system (car, robot, etc.) or to help human beings in their daily tasks.

[0008] One of our concern here is to focus on the data structure for images and its consequences on the processing scheme.

[0009] Algebraic topology concepts such as chain, boundary, cochain and coboundary are a key to representing images. Algebraic topology is well-known domain in mathematics [10, 6, 9]. The literature of algebraic topology offer various knowledge that can be used for images. The specialists for algebraic topology, however, have made no effort to implement their knowledge into computer vision and image processing. Instead of using algebraic topology, the specialists on image processing and computer vision have limited their selves to develop solutions based on the topology and on discrete geometry [7, 8].

[0010] Algebraic topology was first introduced into image processing and computer vision by Allili and Ziou for topological feature extraction and shape representation in binary images [1, 2, 15]. It is used by Allili, Mischaikow and Tannenbaum [3] in medical binary images. Auclair et al. [4] used algebraic topology for linear and non-linear isotropic diffusion as well as for optical flow in gray level images. P. Poulin et al. [12] used algebraic topology for snakes and elastic matching in gray level images.

[0011] In image processing and computer vision, several image models have been accepted and are in recurrent use since several decades. These image models integrate both the image support and the local quantities associated with this support. The image support is formed by pixels. With each pixel is associated a scalar quantity called a gray level, or a vector quantity called either color at the perceptual level or multispectral at the signal level. Existing models differ primarily in terms of how the image support (definition and the organization of pixels) and how values are formulated. The well-known image models are the function, the random process, and the ordered set. The image is a function $L_x \times L_y \to G^m$, where $L_x = \{1,\ldots,N_x\}$ and $L_y = \{1,\ldots,N_y\}$, $N_x \times N_y$ is the resolution of the image, G = {1, ... ,n}, where n is the maximal quantity and m the number of image bands. In the case of a binary image n = 2, image processing has roots in graph theory, language theory, logic, discrete geometry, and so on. If n > 1, usually the image is modeled as a real function (analog image where $G, L_x, L_y \subset \Re$). In this case, function theory, functional analysis, catastrophe theory, differential equations, and differential geometry are the foundation. An image can also be modelled as a collection of random variables $X(i,j) \mid (i,j) \in L_x \times L_y$. In this case, the probability density function, moments, sufficient statistics, time series, and the Markov processes are the roots. When the image is modelled as an ordered set, discrete mathematics and mathematical morphology are the foundation.

[0012] Since the introduction of mathematical morphology in sixteen, efforts of researchers in these fields are focused on the use of more and more complex mathematical, physical or computer concepts as formalism of specific problems (edge detection, image segmentation, optical flow, and defomation) without questioning the image model. The definition of new image model can leads algorithms that are designed on new basis. An image is a physical or mathematical quantity where variables (image support) represent geometrical or temporal elements such as points, lines, surfaces, and times. For example, the image, as a function $L_x \times L_y \to G^m$, can be defined by both the geometrical and topological properties of the domains $L_x \times L_y$ and the topological, geometrical and analytical properties of G^m. Although existing image models have deep roots in mathematics or in physics, the variables, the quantity and the association between them are not well-defined. For a given computer vision or image processing task, no formal mechanism is given for the integration of physical, topological, geometrical properties of objects and their behaviors as a part of the image model. Consequently, the resulting computational schemes are non-modular and sometimes not easy to reproduce.

OBJECTS OF THE INVENTION

[0013] An object of the present invention is therefore to provide an image model based on n-pixels.

[0014] Other objects, advantages and features of the present invention will become more apparent upon reading of the following non-restrictive description of preferred embodiments thereof, given by way of example only with reference to the accompanying drawings.

[0015] It is to be noted that various documents are referred to herein. These documents are hereby incorporated by reference in their entirety.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] In the appended drawings:

[0017] Figure 1 illustrates a 2-cube; the orientation is given by its definition [0,1] X [0,1];

[0018] Figure 2 illustrates a) a subdivision that is a cubical complex and b) a subdivision that is not a cubical complex;

[0019] Figure 3 illustrates, in solid lines, a primary cubical complex and, in dashed lines, a secondary cubical complex;

[0020] Figure 4 illustrates a) an original image and b) the resulting smoothed image;

[0021] Figure 5 illustrates a body in 3D space at time t. a) $\mathbf{x}_1$ is the location of the particle $\mathbf{X}_1$, **n** is a vector normal to the surface, dS is an infinitesimal surface patch and dV is an infinitesimal amount of volume. b) $\mathbf{f}_e$ is an external force applied on dS, p is the mass and **b** is a body force applied on dV. c) **q** is the heat flow passing through dS and r is the heat produced by dV;

[0022] Figure 6 illustrates three examples of q-pixels in $\mathbb{R}^2$ ($I_1 \times I_2$) a) 0-pixel : $I_1 = \{2\}$, $I_2 = \{1\}$ b) 1-pixel : $I_1 = [2,3]$, $I_2 = \{1\}$ c) 2-pixel : $I_1 = [2,3]$, $I_2 = [1,2]$;

[0023] Figure 7 illustrates an example of the coboundary operation;

[0024] Figure 8 illlustrates a projection onto (a) the tangential part and (b) the normal part of the domain;

[0025] Figure 9 illustrates examples of cochains for one 3-pixel of K^{p1} . a) cochain T b) cochain H ;

[0026] Figure 10 illustrates examples of cochains a) Q and b) D for one 3-pixel of K^s;

[0027] Figure 11 illustrates examples of cochains a) T and b) G for one 3-pixel of K^p;

[0028] Figure 12 illustrates three different paths between two points;

[0029] Figure 13 illustrates computational scheme for the unsteady problem with no source;

[0030] Figure 14 illustrates two cubical complexes for a 5 X 5 image;

[0031] Figure 15 illustrates γ_P , one 2-pixel of K^p;

[0032] Figure 16 illustrates a) γ_E (in dashed lines) one 2-pixel of K^3 intersecting four pixels of K^p; b) Cochains T and G c) Cochain Q;

[0033] Figure 17 illustrates a) One 3-pixel of $K^{s'}$ surrounding one 1-pixel of $K^{p'}$;

[0034] Figure 18 illustrates λs for one 2-pixel of K^p;

[0035] Figure 19 illustrates one 3-pixel of K^s intersecting four 3-pixels of K^p. a) Cochain T; b) Cochain G; c) Cochain Q;

[0036] Figure 20 illustrates two 2-pixels of K^s with λ's;

[0037] Figure 21 illustrates a) Physics-based linear isotropic diffusion, $\sigma = \{2.0, 4.0, 5.0\}$; b) Convolved, for same scales;

[0038] Figure 22 illustrates first images of three sequences. (a) Rotating sphere sequence; (b) Hamburg taxi sequence; and (c) tree sequence;

[0039] Figure 23 illustrates a flow pattern computed for the rotating for the rotating sphere sequence, using a) physics-based method and b) iterative finite difference method;

[0040] Figure 24 illustrates a flow pattern computed for the Hamburg taxi sequence, using a) physics-based method and b) iterative finite difference method;

[0041] Figure 25 illustrates a flow pattern computed for the tree sequence, using a) physics-based method and b) iterative finite difference method;

[0042] Figure 26 illustrates a first image of tree sequence with white noise added (standard deviation of 10);

[0043] Figure 27: Flow pattern computed for the tree sequence with white noise added, using a) physics-based method and b) iterative finite difference method;

[0044] Figure 28 illustrates a section of the peppers image ($\sigma = 5$); a) Original image with noise added; b) PB method; and c) FD method;

[0045] Figure 29 illustrates a section of the peppers image ($\sigma = 5$); a) Original image; b) Original with white noise added; c) PB method; d) FD method;

[0046] Figure 30 illustrates: a) PB, $\sigma = \{1.0, 3.0, 5.0\}$; b) FD; for same scales;

[0047] Figure 31 illustrates: a) Original image; b) With noise added (std dev.=10);

[0048] Figure 32 illustrates: a) PB, $\sigma = \{4.0, 8.0\}$; b) FD; for same scales;

[0049] Figure 33 illustrates: a) PB and (b) FD methods with $\sigma = 8.0$;

[0050] Figure 34 illustrates a body of arbitrary size, shape and material. ΔS is a surface patch, **f** is a vector of surface forces applied on ΔS,

ΔV is an amount of volume with a mass ρ, **b** is a vector of body forces applied on ΔV;

[0051] Figure 35 illustrates a cutting plane passing through a point and partitioning the body into two sections;

[0052] Figure 36 illustrates a force Δf acting on a cutting plane with normal vector **n**;

[0053] Figure 37 shows examples of stresses on a body : (a) Original body; (b) Normal stress after an extension of the body; (c) Normal stress after a compression of the body; and (d) Shear stress after a distortion of the body;

[0054] Figure 38 illustrates the component of the stress in the direction of x_i;

[0055] Figure 39 shows the deformation (a) and distortion (b) of a body subjected to stresses; in both figures, the rectangle ABCD has been deformed or sheared into A'B'CD;

[0056] Figure 40 illustrates normal strain of a body;

[0057] Figure 41 illustrates shear strain in a body;

[0058] Figure 42: A body B in motion subjected to forces; t_i^n and pb_i are respectively the traction and body forces in the direction of x_i;

[0059] Figure 43 illustrates (a) the kinematic equation; (b) the constitutive equation; and (c) the conservation equation;

[0060] Figure 44 illustrates the decomposition of the linear elasticity problem into basic laws;

[0061] Figure 45 shows three examples of q-pixels in $\mathbb{R}^2$ (I_1 x I_2) a) 0-pixel : $I_1 = \{2\}$, $I_2 = \{1\}$ b) 1-pixel : $I_1 = [2,3]$, $I_2 = \{1\}$ c) 2-pixel : $I_1 = [2,3]$, $I_2 = [1,2]$;

[0062] Figure 46 is an example of the coboundary operation;

[0063] Figure 47 illustrates examples of cochains for a 3-pixel of K^p; (a) Cochain $\mathcal{U}$ (b) Cochain D;

[0064] Figure 48 illustrates examples of cochains S (a) and F (b) for a 3-pixel of K^p;

[0065] Figure 49 shows two cubical complexes for a 5 x 5 image;

[0066] Figure 50 illustrates a 2-pixel of K^p and the topological quantities associated with it;

[0067] Figure 51 illustrates: (a) γ_F in dashed lines (b) 2-cochains D and $\mathcal{U}$ (c) 2-cochain S;

[0068] Figure 52 illustrates five adjacent vertices in a curve and its deformation;

[0069] Figure 53 is a table that shows typical values of the Poisson's ratio and Young's modulus of elasticity for some materials;

[0070] Figure 54 shows: (a) Initial curve (b) Bright line plausibility image;

[0071] Figure 55 is the results for an aerial image; (a) PB method (b) FEM method;

[0072] Figure 56 is an initial curve for a SAR image;

[0073] Figure 57 shows line plausibility for a SAR image;

[0074] Figure 58 illustrates a road correction for a SAR image with PB method;

[0075] Figure 59 illustrates a road correction for a SAR image with FEM method;

[0076] Figure 60 shows an initial curve;

[0077] Figure 61 shows a bright line plausibility image;

[0078] Figure 62 illustrates corrections for a multiband image (PB method);

[0079] Figure 63 illustrates corrections for a Landsat 7 image (FEM);

[0080] Figure 64 illustrates initial (a) and corrected (b) curves for a synthetic image; and

[0081] Figure 65 (a) through (c): shape recovery of a curve when the external forces are removed. (d) Both initial curve (in white) and final curve (in black).

DESCRIPTION OF THE EMBODIMENT

[0082] An aspect of the present invention is generally based on a data structure for images based on n-pixels, in which the image support, the image quantity and the allowable operations are specified separately. In this data structure, mathematics and physics are unified; that is, the data structure allows taking into account constraints originating in physics, mathematics, and the future use of the image. The image dimension is explicit, which allows us to design algorithms that operate in any dimension. Other advantages and new open problems of this data structure are discussed hereinbelow.

[0083] One of the goals of the present invention is to give a computational image model or a data structure that should be capable of capturing all objects properties that are needed for the given task. A data structure of the image is the formal specification of the image variables, image quantity, the association between quantities and variables that allows to capture the geometrical and topological properties of objects as well as their physical and mathematical behavior. The abstract view of image as a data structure as is used in computer programs is defined by the attributes and a collection of meaningful operations. The attributes are the image support and quantities that are assigned to the image support such as the image radiometry (e.g., color and grey level) or any feature that can be deduced from the radiometry (e.g. texture). These quantities are scalar, vector or tensor. The allowable operations are of two kinds: the operations that are problem independent such as read and write and those that are problem defendant such as object deformation, diffusion and optical flow. To summarize, the image is defined by the support, quantities and allowable operations. We will explain in details each of these three items hereinbelow.

Image support

[0084] Often, the discretization of an image is achieved via a cubic tessellation. For example, a 2D image support is formed by unit squares commonly called pixels. Similarly, a 3D image is a tessellation of unit cubes commonly called voxels. More generally, the present invention will consider the image in n dimensions as a set of unit n-cubes, which we will name n-pixels. When n = 0, the image is set of points; when n = 1, a set of edges; when n = 2, a set of squares, and so on. Any two n-pixels are either disjoint or intersect through a common i-pixel, where i < n. This subdivision of the image support is not unique. Several other geometrical forms such as, for example, triangles or hexagons can be used. It has been proven that the topological features of the image support do not depend on the subdivision used [13]. The cubical subdivision is commonly used in image processing and computer vision and will therefore be used herein. One does not need to explicitly define an orientation on pixels. Indeed, the definition of a pixel as a product of intervals in a certain order coupled with the natural order of real numbers imposes an orientation on each coordinate axis and also on the canonical basis of $\Re^n$ (see Figure 1).

[0085] Formally, a pixel $\sigma \in \Re^n$ as a geometric entity is translated into an algebraic structure as follows:

$$\sigma = I_1 \times I_2 \times \cdots \times I_n \tag{1}$$

where × is the Cartesian product and I_i is either a singleton or an interval of unit length with integer endpoints; i.e., I_i is either the singleton $\{k\}$, in which case it is said to be a degenerate interval, or the closed interval $[k,k+1]$ for some integer k. The number $q \in \{0,1,...,n\}$ of non-degenerate intervals in equation 1 is by definition the dimension of σ, which will be referred to as a q-pixel. For $q \geq 1$, let $j = \{k_0, k_1, ..., k_{q-1}\}$ be the ordered subset $\{1,2,...,n\}$ of indices for which $I_{k_j} = [a_j, b_j]$ is non-degenerate. Define

$$A_{k_j}\sigma = I_1 \times \ldots \times I_{k_j-1} \times \{a_j\} \times I_{k_j+1} \times \ldots \times I_n$$

and

$$B_{k_j}\sigma = I_1 \times \ldots \times I_{k_j-1} \times \{b_j\} \times I_{k_j+1} \times \ldots \times I_n$$

$A_{k_j}\sigma$ and $B_{k_j}\sigma$ are, respectively, the front (q-1)-face and the back (q-1)-face of σ. Each of these (q-1)-faces is a (q-1)-pixel. These faces are then called (q-1)-faces of σ. In the same manner, one can define the (q-2)-faces, ... , down to the 0-faces of σ. In Figure 1, we see a 2-pixel A, with its 1-faces a, b, c, d. The 0-faces of A are the vertices that are not represented here for the sake of clarity of the picture. The boundary of the q-pixel σ allows us to write the relationship between a q-pixel and its (q-1)-faces in algebraic form. It is by definition the alternating sum of its (q-1)-faces, i.e.

$$\partial_q \sigma = \sum_{i=0}^{q-1} (-1)^i (B_{k_i}\sigma - A_{k_i}\sigma) \qquad (2)$$

[0086] For example, the boundary of A in Figure 1 is $\partial_2\sigma = (-0\times[0,1] + 1\times[0,1]) - ([0,1]\times 0 - [0,1]\times 1) = (c-a)+(d-b)$.

[0087] So far, the pixel, its faces, and the association between them have been defined geometrically and algebraically. We will now define the image support as a geometrical entity, called a cubical complex, and then write its algebraic structure. A cubical complex K in $\Re^n$ is a collection of q-pixels where $0 \le q \le n$ such that :

1. Every face of a pixel in K is also in K.
2. The intersection of any two pixels of K is either empty or a face of each of them.

[0088] The first condition implies that all faces of a pixel belong to the cubical complex.

[0089] The second condition concerns the organization of the cubical complex. If the intersection of two pixels is empty, then the image support is formed by one or more connected components. This condition provides an image support that is more general than existing definitions since it allows a formal specification of a cubical complex that is formed either by one or more image supports (e.g., an image sequence) or by several distinct binary objects. When the intersection is a face, certain geometrical configurations of the complex are ruled out. For example, Figure 2a illustrates a subdivision that is a cubical complex and Figure 2b illustrates a subdivision that is not a cubical complex.

[0090] The dimension of K is by definition the largest number q for which K contains a q-pixel.

[0091] As in the case of the q-pixel, the cubical complex can be written in algebraic form. Given a topological space $X \subset \Re^n$ in terms of a cubical complex, we denote $E_q = \{\sigma_q^1, ..., \sigma_q^{N_q}\}$, the set of all q-pixels of X. A q-chain in X is a formal sum of integer multiples of elements of E_q. More precisely, it is a linear combination

$$\lambda_1 \sigma_q^1 + \cdots + \lambda_{N_q} \sigma_q^{N_q} \tag{3}$$

where $\lambda_1, ..., \lambda_{N_q}$ are integers. For example, in Figure 1, $a - c + d - b$ is a 1-chain. Two q-chains are added by adding corresponding coefficients. The set of q-chains can be given the structure of a free Abelian group with basis E_q, usually denoted by $C_q(X)$. Since we will only be concerned with finite complexes, the groups $C_q(X)$ are finitely generated and $C_q(X) = 0$ if q is greater than the dimension of the complex; naturally, $C_q(X) = 0$ if q < 0.

[0092] To define the chains that are associated with the faces of pixels of a cubical complex, we extend equation 2 by linearity to all q-chains and obtain the boundary map $\partial_q : C_q(X) \to C_{q-1}(X)$. Note that $\partial_0 = 0$ since $C_{-1}(X) = 0$. The boundary map satisfies the following very important property [9]:

$$\partial_q \circ \partial_{q+1} = 0 \tag{4}$$

[0093] To summarize, the discrete image support of any dimension n is formed by n-pixels. Unlike conventional image models, the n-pixel is a dimensional geometric entity formed by other geometric entities called faces. The geometrical n-pixel is translated into a recurrent algebraic structure, more reliable for mathematical handling. All n-pixels of the image support form a cubical complex, a geometrical entity that is translated into an algebraic structure called the chain. The association between the n-pixel and its faces, and hence between chains of successive dimensions is given by the boundary operator. It is to be noted that the use of any other geometrical primitive such as the triangle or hexagon for subdividing the image support affects the computational complexity of the derived algorithms, but it affects neither the topological features of the image support nor the computation rules for these features.

Image Quantities

[0094] We introduced hereinabove the concept of the finite cubical complex to give an algebraic description of the discrete image support. A similar formulation is required to describe the image field (scalar, vector, matrix) over the discrete image support. For this purpose, we return to the chain concept described above and give a more general definition. Let us consider a cubical complex K of dimension n. We associate with each q-pixel ($q \leq n$) of K a coefficient in the ring (A,+,*), where the elements may be scalars (gray level), vectors (color, gradient), matrices (Hessian), etc. The chain is the formal sum $\sum_i \lambda_i c_q^i$, where λ_i is a coefficient in (A,+,*), and the generators c_q^i, $\forall i$ form a basis of an Abelian group. The chain can be seen as a vector $[\lambda_1,\ldots,\lambda_{N_q}]^T$, where N_q is the number of generators used. Consequently, two chains can be summed by adding their coefficients and multiplied using the scalar product. The addition and multiplication are taken according to the definition of a ring. Moreover, one can define the null chain (resp. unit chain) whose coefficients are all equal to the null (resp. unit) element of the + (resp. *) operation of (A,+,*). Consequently, q-chains define an image model that has attractive computational properties since they form a rich algebraic structure, the module (i.e, a vector space defined on a ring).

[0095] To briefly show how to use the chain model in image processing, we present a simple example concerning any global transform of the image such as histogram equalization or thresholding. The chain coefficients are the gray levels. The formal expression of the global transform implies two applications: $\overline{H}:(\varepsilon_A,+,*)\rightarrow(\varepsilon_A,+,*)$ and $H:(A,+,*)\rightarrow(A,+,*)$, where $(\varepsilon_A,+,*)$ is a module. They are defined by $\overline{H}(\sum_i\lambda_i c_i)=\sum_i H(\lambda_i)c_i$.

[0096] The drawback of the conventional chain-based image model is that the physical or mathematical quantities and the image support are described together in the formal sum. Consequently, the chain coefficients combine mathematical or physical quantities, pixel orientation, and possibly other quantities such as weights associated with pixels. For example, there is ambiguity in the interpretation of the sign of λ_i: it may be due to the orientation, the physical quantities, the weights or their multiplication. This image model can be considered adequate, especially as in physics [14], engineering and computer graphics [11, 5], chains have been used to model quantities. However, we propose to refine this quantity model to overcome the confusion produced by the chain coefficients.

[0097] To reflect only the geometry of the image support (e.g., orientation and multiplicity), in what follows we consider n-chains with integer coefficients. We are looking for an application $F_q:C_q(X)\rightarrow(B,+,*)$, which associates a global quantity (energy, gray level, color, flux, tensor, etc.) with all q-pixels, where $q\leq n$ and $(B,+,*)$ is a ring. As in the case of the chain-based image model, for two adjacent q-pixels c_q^1 and c_q^2, F_q must satisfy $F_q(\lambda_1 c_q^1+\lambda_2 c_q^2)=\lambda_1 F_q(c_q^1)+\lambda_2 F_q(c_q^2)$, which means that the sum of the quantities generated within each q-pixel is equal to the quantities generated within the two q-pixels. F_q can be extended by linearity to any q-chain $\sum_i\lambda_i c_q^i$, where each λ_i is an integer, as follows:

$$F_q(\sum_i \lambda_i c_q^i) = \sum_i \lambda F_q(c_q^i).$$

[0098] F_q is called a q-cochain. To illustrate the cochain concept, let us consider a 2-pixel c$_2$ and a vector field V. A 0-cochain is defined by the value of V at 0-pixels. A 1-cochain is $\int_{\partial c_2} V.ds$, the line integral over the faces of c$_2$. A 2-cochain is $\int_{c_2} V.dS$, the surface integral over the 2-pixel c$_2$, and the "." the dot product.

[0099] To summarize, a cochain allows us to associate quantities with an n-pixel and with its faces. Unlike existing image models, our model provides a rich structure that allows the definition of both local and global quantities.

Operations

[00100] A generic operation that can be instantiated depending on the problem that we are dealing with will now be defined. Having in mind that the q-pixel has 3^q faces, the generic operation should specify the algebraic relationship between the quantities (i.e., cochains) associated with these faces. Based on the linearity principle, the quantity of a given q-pixel is transferred to its cofaces with the same or opposite sign, according to the agreement of its orientation and the orientation of its cofaces. The quantities that are transferred to the (q+1)-pixel by its faces are summed. More formally, it should be recalled that the relationship between the (q-1)-chain and the q-chain is given by the boundary operator. Similarly, the relationship between the q-cochain and the (q+1)-cochain is given by the coboundary operator $\delta_q : C^q \rightarrow C^{q+1}$, where C^q is the Abelian group of q-cochains. Given a (q+1)-chain c, this operator is defined by:

$$\delta_q \mathcal{F}_q(c) = \mathcal{F}_q(\partial_{q+1} c) \qquad (5)$$

[00101] The capacity of the cochain and the coboundary to model a given problem will now be discussed. The cochain is a linear application and should fulfill the coboundary requirement in equation 5. Thus a question concerns the limits in modeling a given quantity. It is difficult to answer this question for the general case. Much investigation is needed first. We only answer by identifying several problems that can be modeled by the cochain and coboundary. Certain problems such as convolution and its applications (smoothing, numerical differentiation, high-pass filtering, noise estimation) and the Fourier transform can be modeled by the cochain without approximation since they only require setting the coefficients $\lambda_i$of the cochain to specific values (see [16] for the case of numerical differentiation). Thresholding can be represented by the cochain without approximation since $H(\sum_i \lambda_i Q_i) = \sum_i \lambda_i H(Q_i)$, where $H:(B,+,*) \to (B,+,*)$. Other problems can be broken down to basic laws, each of which can be described by the topological equation in (equation 5). For example, it has been pointed out [14] that many physics problems can be broken down into basic physical laws such as balance and constitutive laws. As we will show hereinbelow, balance law can be written in a discrete form by using the topological equation in (equation 5) without approximation. The constitutive laws cannot be translated in algebraic form without approximation. Usually, they requires the link between cochains that belong to different cubical complexes. For example, in the case of dual complexes, two cochains are linked by an algebraic linear system. We call this transformation the codual operation. More generally, 0-cochains represent local quantities. However, q-cochains (q >1) represent global quantities (e.g., an integral or the summation of a differential form) since they are associated with the algebraic structure of an edge, an area, a volume, etc. Thus, the cochain, coboundary, and codual are generic algebraic structures that can be instantiated by physical or mathematical laws. The exact translation of given problem in terms of q-cochains is possible if one is able to find the basic laws that can be described without approximation by either cochains or the topological equation in (equation 5).

[00102] In our previous example, we can interpret the coboundary as follows: assuming that the vector field is conservative $V = \Delta v$, we have $\int_{c_1} \Delta v.ds = v(b) - v(a)$ which is a coboundary operator since it may be written as $\delta_0 F_0(c_1) = F_0(\partial_1 c_1)$, where a and b are the faces of c_1. Similarly, $\int_{c_2} div(V)dS = \int_{\partial c_2} V.nds$, where n is the outward unit normal vector to ∂c_2, which is a coboundary operator since it may be written as $\delta_1 F_1(c_2) = F_1(\partial_2 c_2)$. This example shows that the coboundary operator may be an exact discrete representation of the fundamental calculus theorems (line integral and Gauss).

[00103] Thanks to the chain, cochain, coboundary and codual concepts, the image model of the present invention can take into account the mathematical or physical laws related to the application. It can thus be instantiated to the various problems of image processing and computer vision. The underlying computational framework is strongly different form existing frameworks. For example, let us consider physics based problems such as optical flow, diffusion and deformation. Existing frameworks can be summarized as follow: 1) modeling by partial differential equation; 2) resolving the PDE by using numerical analysis scheme or Fourier space.

[00104] The computational framework, according to an aspect of the present invention, can be summarized as follow: 1) identification of its basic laws; 2) the definition of the image support that is the number of cubical complexes and the dimension of the cubes (e.g., for the case of a multi-resolution processing); 3) the definition of global and local quantities; 5) instantiation of the coboundary and codual operators; 6) the resolution of resulting algebraic system. The advantages of this framework are described hereinbelow and will be better defined in two practical examples.

[00105] To summarize, the coboundary operator links quantities associated with the faces of an n-pixel. Codual operator links quantities

associated to complexes of an image. If a given problem can be broken down into basic laws, the cochain and coboundary are the discrete representation of these basic laws. Cochain, coboundary, and codual are generic concepts that can be instantiated by physical or mathematical laws. Thus, they can be used in various computer vision and image processing problems.

An Example

[00106] Let us consider the linear isotropic diffusion in gray level images which is a physics-based problem. The reader can find all details in [4]. For the sake of clarity and without loss of generality, we will limit ourselves to considering the 2D global differential equation for heat flow in a homogeneous medium. Let V be a 3D region with boundary S inside the flow. The rate of heat flow across S out of V is given by:

$$\int\int\int_V \sigma(\mathbf{x},t)dV = -\int\int\int_V \nabla.(\lambda\nabla T(\mathbf{x},t))dV \qquad (6)$$

where **x** is a spatial vector, t time, and λ a positive constant.

[00107] To resolve this problem, the image is defined by a continuous scalar field T, the temperature (i.e., gray level), two dual cubical complexes (i.e., two chains), and three cochains. If only one cubical complex is used, two différent orientations are associated with each 1-face. To overcome this problem, two dual complexes (primary and secondary) are used (see Figure 3). Concerning the use of three cochains, it has been pointed out in [4] that this EDP is formed by three basic physical laws. Each cochain is associated with one law. The first is the thermal tension law (also known as Fick's Law), which states that heat flows from regions of higher temperature towards regions of lower temperature. The direction of the gradient ∇T is the direction of the largest increase in temperature; the heat flows in the opposite direction. Formally, this law is written as follows:

$$g(\mathbf{x},t) = -\nabla \mathcal{T}(\mathbf{x},t) \tag{7}$$

[00108] The primary cubical complex is the support for this balance law. The orientation plotted on this cubical complex is the direction of the path on which the integral is computed. Let us assume that the 0-cochain is the temperature T associated with 0-pixel c_0. The 1-cochain G associated with 1-pixel c_1 is the global thermal tension transferred by the two 0-pixels that are the faces of c_1. Consequently, the topological equation is:

$$\mathcal{G}(c_1) = \delta_0 \mathcal{T}(c_1) = \mathcal{T}(\partial_1 c_1) \tag{8}$$

[00109] By the linearity of cochains, this topological equation is valid for all 1-chains.

[00110] The second law, called the heat source law, concerns the net outflow of internal thermal energy at the point **x** and time t. This is a balance law. It is given by:

$$\sigma(\mathbf{x},t) = \nabla.q(\mathbf{x},t) \tag{9}$$

[00111] When ∇.q(x,t) > 0, the outflow is positive and thermal energy must flow away from **x**. Similarly, if ∇.q(x,t) < 0, the inflow is larger than the outflow and thermal energy increases at **x**. The equilibrium for a diffusion process is attained if ∇.q(x,t) = 0. Let us consider the secondary cubical complex. The orientation plotted on this cubical complex is the direction of the flow. The 2-cochain Σ associated with the 2-pixel c_2 is the global heat transferred by the faces of c_2

$$\Sigma(c_2) = \delta_1 \mathcal{Q}(c_2) = \mathcal{Q}(\partial_2 c_2) \tag{10}$$

[00112] This topological equation is valid for all 2-chains. The third law is constitutive (it depends on the medium feature). It makes the link between the flow density law and the heat source law. It is given by:

$$q(x,t) = \lambda g(x,t) \tag{11}$$

[00113] This equation cannot be discretized exactly, since it involves global quantities defined on two dual complexes. Consequently, the 2-cochain ± cannot be computed without approximation from the 1-cochain G. For example, they can be linked as a linear equation system $\Lambda Q = \pm$, where Λ is the coefficient matrix. Figure 4 gives an original image and the image smoothed using this computational scheme.

[00114] The data structure associated to the linear diffusion problem is defined by: 1) two dual cubical complexes; 2) two cochains G and Λ for global quantities and a scalar field T; 3) two coboundary operations for balance laws and an codual operation that represent the constitutive law. The framework is summarized as follows: g is approximated by a bilinear polynomial. The cochain G is computed using a line integral. q is computed using the constitutive law in equation 11. The cochain ± is computed using Gauss's theorem. Finally, a system of linear equations obtained from equation 11 is obtained where the unknowns are T. It should be noted that the cochains in equations 8 and 10 are computed without approximation.

[00115] The image model according to an aspect of the present invention and described hereinabove may generally be characterized by three major points: 1) the image support and quantities are defined separately and then linked together via algebraic language; 2) the pixel is dimensional and is written in an algebraic form; 3) both local and global quantities are represented by the cochains and coboundary operator.

[00116] Each of these specificities will now be discussed to show their straightforward consequences for image processing.

[00117] The separability of the image model allows the distinction between image variables and image quantities. The image variables offers numerous possibilities for existing mathematical formulations such as the use of algebraic topology to help in the design of algorithms. For example, binary image algorithms are written as algebraic systems [16]. The well defined quantities allow the use of physics, vector analysis or differential forms in the design of algorithms. Taking image support and image quantities together, well-known problems such as those of diffusion and optical flow in gray level images can be written as algebraic systems. Furthermore, the transfer of quantities between a given domain and its boundary is straightforward, using the concepts of cochain and coboundary as a general framework. For example, as we have shown, in vector calculus, this transfer is easier thanks to the three fundamental calculus theorems, the line integral, Stokes's theorem, and Gauss's theorem.

[00118] Unlike existing image models, by considering the pixel as dimensional primitive the connectivity paradox of the image support is avoided [8]. That is, the well-known Jordan theorem is fulfilled. Its decomposition into faces and the use of cubical structures such as chains make the dimension of the image explicit. Algorithms designed according to this formalism operate in any dimension. This fact overcomes the traditional limitations that we face in designing an algorithm, say in one dimension, extending it to two dimensions, and then to three dimensions, and so on. Each extension step may be a difficult task.

[00119] The definition of the cochain depends on the problem that we are dealing with. Thus, this image model offers real flexibility for the integration of mathematical objects (scalar, vector, tensor) and physical laws (balance, constitutive). Furthermore, the use of global quantities associated with an n-

pixel implies noise reduction. In fact, global quantities are computed from the field by using the integral or the discrete summation. As the opposite operation from the differentiation, which enhances high frequencies of the image, the integral performs a smoothing operation. It allows us to reduce the order of the derivative used in an image-processing scheme. Consequently, the introduction of global quantities may allow the use of higher-order derivatives.

[00120] Another contribution of the image model according to an aspect of the present invention concerns the numerical scheme used to solve nonlinear problems such as the diffusion problem and elastic matching. It should be recalled that, usually, a problem is formulated and then a numerical analysis scheme is used. The numerical analysis scheme may not have been derived for exactly this formulation and many approximations must be made. The explanation of intermediate results is not available. Consequently, no clear idea is available about the convergence of the numerical analysis scheme and the numerical results obtained may be a broad approximation of the desired solution. Based on the problems tackled, we conclude that in the image model presented here, the numerical scheme is deduced from the problem model with little or no approximation [4, 12]. In fact, various problems may be broken down into basic laws and then reformulated in terms of cochains and coboundaries. Thus they can be written as linear algebraic systems and solved.

PRACTICAL EXAMPLE #1:

PHYSICS-BASED RESOLUTION OF DIFFUSION AND OPTICAL FLOW

[00121] We will now describe an alternative to partial differential equations (PDEs) for the solution of three problems in computer vision: linear isotropic diffusion, optical flow and nonlinear diffusion. These three problems are modeled using the heat transfer problem. Traditionally, the method for solving physics-based problems such as heat transfer is to discretize and solve a PDE by a purely mathematical process. Instead of using the PDE, we propose to decompose the global heat problem into basic laws. We show that

some of the basic laws admit an exact global version since they arise from conservation principles. We also show that the assumptions made on the other basic laws can be made wisely, taking into account knowledge about the problem and the domain. We use the above-described image model which allows us to encode physical laws by linking a global value on a domain with values on its boundary. The resulting algorithm performs in any dimension. The numerical scheme is derived in a straightforward way from the problem modeled. It thus provides a physical explanation of each solving step in the solution.

Background

[00122] In recent years, partial differential equations (PDEs) have attracted increasing interest in the field of computer vision. Since PDEs have been the subject of much study by numericians, powerful numerical schemes have been developed to solve them. Consequently, domains such as image enhancement, restoration, multi-scale analysis and surface evolution all benefit greatly from PDEs [25].

[00123] One important class of equations governing certain physical processes is the linear elliptic PDE of the general form known as the Helmholz equation:

$$\nabla^2 u(\mathbf{x}) + p(\mathbf{x})u(\mathbf{x}) = f(\mathbf{x}) \qquad (12)$$

where **x** denotes a vector in the n-dimensional space, u(**x**) is the dependent variable, ∇^2 is the Laplacian operator, and p(**x**) and f(**x**) are spatial functions. When p(**x**) = 0, we have the Poisson equation [21, 32] (also known as the non-homogeneous Laplace equation [30, 44]). One of the physical processes governed by Equation 12 is steady-state heat transfer.

[00124] In the field of computer vision, Equation 12 may arise from two approaches. The first is variational calculus. As a matter of fact, many

problems such as shape from shading [39], surface reconstruction [32, 40] and the computation of optical flow [29] can be formulated as variational problems. The solutions to these variational problems are given by Euler-Lagrange equations, which are in the form of Equation 12 [31, 32]. The second approach is physics-based. For example, diffusion processes arise from heat equations and shock filters from work in fluid mechanics [25]. For both the variational and the physics-based approaches, the resulting PDEs are continuous and have to be discretized.

[00125] Traditionally, the discretization of PDEs in computer vision has been done by applying finite difference methods [23, 31, 39, 40]. Equation 12 is solved iteratively using either a direct Fourier-based Poisson solver for each iteration [39], finite elements [24], or spectral methods [32]. Iterative methods such as those in [39] do not ensure convergence unless smoothness is very high [21].

[00126] We can summarize the existing methods for the resolution of problems involving PDEs as follows: 1) identification of basic laws; 2) combination of the basic laws in order to write the PDEs; 3) discretization of the PDEs; 4) resolution of the PDEs via a numerical method.

[00127] This process, which has been used in various fields of application, is purely mathematical. Consequently, it has the following drawbacks: 1) Some quantities involved in the solution process do not have a physical interpretation; 2) This lack of interpretation is manifested in intermediate solutions involving iterative processes and since these solutions cannot be physically explained, discovery of the optimal solution cannot be ensured in an optimal time.

Solution

[00128] To overcome these drawbacks, an alternative to PDE resolution in the context of the heat transfer problem is proposed and will be described hereinbelow.

[00129] Generally, basic laws in physics-based problems are combined into a global conservation equation [42] that is valid on the whole body or a part of it. A local conservation equation (PDE) is then obtained by considering the particular case of a part of a body reduced to a point.

[00130] In discrete problems such as those encountered in computer vision, the continuous domain is subdivided into many sub-domains in which there is only one value available, which can be considered as a global value. Therefore, instead of using the PDE, we propose to use the global conservation equation directly on each sub-domain.

[00131] In order to handle these physical laws which link global values at points, lines, surfaces, volumes, etc. we use the image model with roots in computational algebraic topology described hereinabove. This model makes it possible to represent global values on entities of any dimension at the same time.

[00132] The above described methodology presents a number of advantages:

1) Many of the basic laws arise out of conservation principles and hence they are valid either at a point (local form as in Equation 12) or over an entire region (global form). Fundamental theorems of calculus such as the Gauss, Green and Line Integral theorems allow the computation of the coboundary operator without any approximation.

2) Some laws require approximations that can be performed wisely, taking into account knowledge about the problem and the domain.

3) The intermediate results have a physical explanation because they represent physical quantities. For that reason, every step has a physical interpretation. Thus we no longer have problems of non-optimality of the solution, because we avoid non-temporal iterative methods.

4) As mentioned earlier, this method can be used with other physics based problems by applying the appropriate basic laws [36].

5) Thanks to the image model, the resulting algorithm performs in any dimension.

6) The computational rules associated with the coboundary operator can be changed without changing the formalism of the operation itself.

7) The same formalism can be used for pixel-based and other types of decomposition of the image (e.g. regions).

[00133] In order to validate the method, we resolve the equation for steady state heat transfer in two applications: the linear diffusion and optical flow. These problems generate equations of the form of Equation 12 or its global version. We also use our methodology to resolve unsteady heat transfer with no source and we apply it to non-linear diffusion.

Physical Principles (explanation of the physical foundations of the heat transfer problem)

[00134] We distinguish two interesting particular cases for diffusion and optical flow problems: steady-state heat transfer and unsteady heat transfer with no source.

[00135] Two important classes of laws are present: conservation and constitutive laws. Conservation laws are independent of the properties of the material, whereas constitutive laws are specific to them. The physical

properties associated with a moving object are energy, work and heat. In what follows, we describe each of these quantities.

Energy Modeling

[00136] Let us define some quantities for a continuous body occupying a volume V bordered by a surface S in a 3D space. We can say that such a body is composed of infinitely many particles (as many as desired), particles being the smallest elements [33]. Figure 5(a) illustrates such a body. At time t, a particle labeled **X** occupies a specific position:

$$\mathbf{x} = (x(t), y(t), z(t))$$

[00137] Each particle can move in space, so a velocity vector is associated with it at time t:

$$\mathbf{v}^*(\mathbf{X}, t) = \frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}, t)$$

[00138] Physical quantities can be associated with a particle labeled **X** (material description) or a position **x** in space (spatial description). For example, **v***(**X**,t) is the material description of the velocity of particle **X** and **v**(**x**,t) is the spatial description of the particle located at position **x**. For our purpose, we use spatial descriptions to derive the heat transfer equation. Vector **n**(**x**,t) is the outward direction of the surface at point **x**.

[00139] The mass Δ m of a small amount of volume ΔV of a body is a measure of its inertia (tendency to resist motion). We use the term mass density, ρ to denote the following quantity:

$$\rho = \lim_{\Delta V \to 0} \frac{\Delta m}{\Delta V}$$

[00140] ρ (x,t) is thus the mass density of the particle located at **x** at time t.

[00141] Two kinds of energy are associated with a moving object: kinetic and internal energy. Kinetic energy is a measure of the state of motion of a body: the faster the body moves, the greater its kinetic energy [28]. Because it is a measure of inertia, kinetic energy also takes the mass into account. For a particle located at **x** at time t, the kinetic energy is thus defined as

$$K(\mathbf{x},t) = \frac{1}{2}\rho(\mathbf{x},t)\,(\mathbf{v}(\mathbf{x},t)\cdot\mathbf{v}(\mathbf{x},t))$$

where . is the dot product. To obtain the kinetic energy for the entire body at time t, K(**x**,t) is integrated over the volume V:

$$K(t) = \frac{1}{2}\iiint_V \rho(\mathbf{x},t)\;(\mathbf{v}(\mathbf{x},t)\cdot\mathbf{v}(\mathbf{x},t))\;dV$$

where dV is an infinitesimal amount of the volume V.

[00142] Internal energy is a measure of the state of temperature of a body. The hotter the body, the greater its internal energy. At time t, each particle has an internal energy density ε (**x**,t) associated with it. The internal energy density is proportional to the temperature of the particle T(**x**,t) with a material-specific heat constant c; that is, ε (**x**,t) = cT(**x**,t). For the entire body, the total internal energy is integrated over the volume V:

$$E(t) = \iiint_V \rho(\mathbf{x},t)\;\varepsilon(\mathbf{x},t)\,dV \tag{13}$$

[00143] The total energy for the body can now be defined as K(t) + E(t).

Work Modeling

[00144] Let us suppose that a body is submitted to an external force $\mathbf{f}_e$(x,t) (e.g. a traction force) and an internal density force **b**(x,t) (e.g. gravity). Figure 5(b) presents the action of external and internal forces. We define work as energy transferred to a body by means of a force acting on the body [28]. Work is negative when the energy is transferred from the body. Suppose that **F**(x) is an internal or external force that is constant over time, acting on a particle located at **x** during an amount of time t. This force will produce a displacement of the particle to position $\mathbf{x}_1$. This displacement is $\Delta\mathbf{x} = \mathbf{x}_1 - \mathbf{x}$. The work w(**F**, **x**) done by this force during this time is

$$w(\mathbf{F},\mathbf{x}) = \mathbf{F}(\mathbf{x}) \cdot \Delta\mathbf{x}$$

and the instantaneous power P(**x**, t) is:

$$P(\mathbf{x},t) = \lim_{\Delta t \to 0} \frac{w(\mathbf{F},\mathbf{x})}{\Delta t} = \mathbf{F}(\mathbf{x}) \cdot \lim_{\Delta t \to 0} \frac{\Delta\mathbf{x}}{\Delta t} = \mathbf{F}(\mathbf{x}) \cdot \frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x},t)$$

[00145] External forces act essentially on the surface of the body. The instantaneous work P_e(t) done by the external forces on the entire body is thus the result of the integration of the external power over the surface:

$$P_e(t) = \iint_S \mathbf{f}_e(\mathbf{x},t) \cdot \mathbf{v}(\mathbf{x},t)\, dS$$

where dS is an infinitesimal part of the surface of the body.

[00146] Defining **b**(**x**, t) as the internal density force, the internal force is thus ρ(**x**, t)**b**(**x**, t). The rate of work over time done by the internal forces on the entire body is obtained by integrating internal work over the volume:

$$P_i(t) = \iiint_V \rho(\mathbf{x}, t)\,(\mathbf{b}(\mathbf{x}, t) \cdot \mathbf{v}(\mathbf{x}, t))\,dV$$

The total work is thus P(t) = P_e(t) + P_i(t)

Heat Modeling

[00147] Heat can be defined as energy transferred to a body owing to a difference in temperature. The heat flow density vector **q(x, t)** is a measure of the rate of heat conducted into the body per unit area per unit of time. We will see later how **q(x, t)** is defined. The external heat addition rate over time is the amount of heat coming from outside the body and entering by its surface. It is computed by projecting **q(x, t)** onto the inward normal vector (–**n(x, t)**) and integrating this projection over the surface:

$$Q_e(t) = \iint_S \mathbf{q}(\mathbf{x}, t) \cdot (-\mathbf{n}(\mathbf{x}, t))\,dS \tag{14}$$

[00148] Now if a body has a rate of heat generation per unit of volume and time **r(x, t)**, the internal rate of heat addition over time is computed by integrating **r(x, t)** over the volume:

$$Q_i(t) = \iiint_V \rho(\mathbf{x}, t)\,r(\mathbf{x}, t)\,dV$$

[00149] Figure 5(c) shows **q(x, t)** and **r(x, t)**. To simplify, we define the source σ **(x, t)** as the rate of heat generated in a particle located at **x** per unit of volume and time:

$$\sigma(\mathbf{x}, t) = \rho(\mathbf{x}, t) r(\mathbf{x}, t) \tag{15}$$

[00150] In many cases, this source is known. However, it could also be a linear function of the temperature: σ(x, t) = a(x, t) + b(x, t)T(x, t) [38]. The total rate of heat addition over time is thus:

$$Q(t) = Q_e(t) + Q_i(t)$$

Energy Conservation Law

[00151] A class of equations in continuum mechanics are those describing the conservation (equilibrium) principles. They express the conservation of certain physical quantities (mass, momentum, energy, etc.) over an entire body and as such they take the form of global equations over the whole body or a part of it [33].

[00152] Conservation principles can be seen intuitively as follows: the change in the total amount of a physical quantity inside a body is equal to the amount of this quantity entering or leaving the body (through the boundary) and the amount generated or absorbed within the body. These laws are applicable for all continuous materials, moving and stationary, deformable and non-deformable, and must always be satisfied. The global conservation equations can then be used to derive their local counterparts, called the associated field equations, which are valid at each point of the body including its borders.

[00153] The first law of thermodynamics, which is relevant for the understanding of the heat transfer equation will now be discussed. This law involves both kinetic and internal energies and states that the total variation of energy in a body (or a part of a body) is the result of the time rate of work and the rate of heat addition combined:

$$\frac{d}{dt}(E(t) + K(t)) = P(t) + Q(t). \qquad (16)$$

[00154] For heat transfer, we are only interested in the case of immobile bodies, that is **v** = (0,0,0), **n**(**x**,t) = **n**(**x**) and ρ(**x**,t) = ρ(**x**). Equation 16 thus becomes

$$\frac{d}{dt}\iiint_V \rho(\mathbf{x})\,\varepsilon(\mathbf{x},t)\,dV = \iint_S -(\mathbf{q}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x},t))\,dS + \iiint_V \sigma(\mathbf{x},t)\,dV$$

which now states that the thermal energy variation in a body is due to internal heat production added to the heat flowing into the body. Using the divergence theorem for Q_e [33] and recalling that ε(**x**,t) = cT(**x**,t), we obtain the thermal energy conservation law:

$$\iiint_V \rho(\mathbf{x})\,c\frac{d}{dt}T(\mathbf{x},t)\,dV = \iiint_V -\nabla\cdot\mathbf{q}(\mathbf{x},t)\,dV + \iiint_V \sigma(\mathbf{x},t)\,dV \qquad (17)$$

where ∇. is the divergence operator. To simplify, let us define the temperature variation h(**x**,t) = $\frac{d}{dt}$T(**x**,t). Equation 17 is a conservation equation and is thus valid over the entire body, a part or a point of this body. Consequently, the integral signs can be taken off:

$$\underbrace{c\,\rho(\mathbf{x})\,h(\mathbf{x},t)}_{\text{Thermal energy variation}} = \underbrace{-\nabla\cdot\mathbf{q}(\mathbf{x},t)}_{\text{Rate of heat entering}} + \underbrace{\sigma(\mathbf{x},t).}_{\text{Rate of heat generation}} \qquad (18)$$

[00155] This equation is said to be local, whereas Equation 17 is said to be global. The thermal energy variation is called the unsteady term, the rate of heat entering is called the diffusion term and the rate of heat generation is called the source term.

[00156] Based on Equation 18, two cases are be considered: the steady state case and the unsteady case with no source. The term steady simply means that there is no variation of the thermal energy of the system over time. That is, the left side of Equation 18 is null:

$$c\rho(\mathbf{x})h(\mathbf{x},t) = 0 \qquad (19)$$

[00157] This implies that the heat diffusion compensates for the internal heat production:

$$\nabla \cdot \mathbf{q}(\mathbf{x},t) = \sigma(\mathbf{x},t) \qquad (20)$$

[00158] In the unsteady case with no source we have

$$\sigma(\mathbf{x},t) = 0 \qquad (21)$$

which means that the time variation of thermal energy is explained by the heat diffusion alone:

$$c\rho(\mathbf{x})h(\mathbf{x},t) = -\nabla \cdot \mathbf{q}(\mathbf{x},t) \qquad (22)$$

Constitutive Principles

[00159] In Equation 18, there are three unknown variables: ρ(**x**), h(**x**,t) and **q**(**x**,t). Let's look at the example of **q**(**x**,t). Suppose that we can measure the time variation of the thermal energy (left side of the equation) and also of the temperature T. We know that **q**(**x**,t) is related to the temperature, but since different materials usually have different diffusion properties, the missing equation **q**(**x**,t) = f(T,**x**,t) must depend on properties of the material we are studying, such as its homogeneity and type of diffusivity. Consequently, the system of equations contains more unknown variables than equations and the function f(T,**x**,t) must be added to the system formed by Equation 18. This is due to the difference in material properties. Different materials behave differently, but are subject to the same conservation laws. Constitutive equations such as f(T,**x**,t), which reflect the internal constitution of materials, allow us to complete the system of equations.

Decomposition into Basic Laws

[00160] We have seen hereinabove that conservation equations are always valid regardless of the materials, whereas constitutive equations are dependent on their properties. When solving directly PDEs like Equation 18 in a discretized context with methods such as the finite differences approach, one makes global assumptions about the time and space behavior of the diffusion, energy variation and source terms without taking into account the nature of the basic laws underlying the problem. Some of these do not require any approximation since they come from conservation principles. Also, a more physically realistic solution can be obtained by choosing a proper approximation for each basic law arising from a constitutive principle. Consequently, we propose to decompose the terms of Equation 18 into basic laws. This equation can be broken down into one constitutive and two conservative laws for the steady state case. In the unsteady case, an additional constitutive and another conservative law must also be considered. Note that since the source term is often known, we do not try to decompose it. Recalling that the diffusion term α (**x**,t) is the rate of heat entering the particle located at **x** at time t, then

$$\alpha(\mathbf{x}, t) = -\nabla \cdot \mathbf{q}(\mathbf{x}, t) \tag{23}$$

is a first basic conservation law.

[00161] The second conservation law concerns the thermal tension. We first define the thermal tension vector **g**(**x**,t) as the vector representing the direction and magnitude of the greatest temperature decrease at a fixed time t. As **g**(**x**,t) is source-oriented (from hot to cold), we must put a minus sign before ∇T(**x**,t) which represents the direction and magnitude of the greatest temperature increase:

$$\mathbf{g}(\mathbf{x}, t) = -\nabla T(\mathbf{x}, t) \tag{24}$$

[00162] This equation is a second basic law. Since the thermal tension is the gradient of a scalar field, it is by definition a conservative field in space. We say that –T(x,t) is the potential field of **g**(x,t) [26, 41].

[00163] The third law is a constitutive law. We define the heat flow density **q**(x,t) as the quantity and the direction of the heat flowing into the particle located at point **x** at time t. It is represented by a vector and greatly depends on the behavior of the material. In the case of uniform, homogeneous materials, it has been proven experimentally by Fourier [20, 27] that **q**(x,t) is directly proportional to the difference of temperature relative to neighbors of this particle:

$$\mathbf{q}(\mathbf{x}, t) = \lambda \mathbf{g}(\mathbf{x}, t) \tag{25}$$

where λ is a material-specific thermal conductivity constant. The value of λ is known for many types of materials. Equation 25 is called the Fourier heat conduction law. For a non-homogeneous material, we consider that it has the behavior of a homogeneous material on an infinitesimal patch, but the conductivity changes with each patch; that is,

$$\mathbf{q}(\mathbf{x}, t) = \lambda(\mathbf{x}, t)\mathbf{g}(\mathbf{x}, t)$$

[00164] For the unsteady case, the fourth basic law is

$$\epsilon(\mathbf{x}, t) = c\rho(\mathbf{x})h(\mathbf{x}, t) \tag{26}$$

where ε (**x**,t) is the thermal energy variation (the unsteady term). This equation is a constitutive one because it involves ρ (**x**), which is material-dependent.

[00165] Finally, the fifth basic law is related to the temperature variation and is a conservation law:

$$h(\mathbf{x},t) = \frac{d}{dt}T(\mathbf{x},t) \tag{27}$$

[00166] Considering only the temperature $T_x(t)$ of the particle located at **x**, we reduce the basic law to a 1-dimensional equation and we can thus say that $T_x(t)$ is a conservative field in time-space.

[00167] To summarize, we have three basic laws for the diffusion term of equation 18, that is:

$$\begin{aligned} \alpha(\mathbf{x},t) &= -\nabla \cdot \mathbf{q}(\mathbf{x},t) \\ \mathbf{q}(\mathbf{x},t) &= \lambda \mathbf{g}(\mathbf{x},t) \\ \mathbf{g}(\mathbf{x},t) &= -\nabla T(\mathbf{x},t) \end{aligned}$$

[00168] We also have two additional basic laws for the unsteady term, that is:

$$\begin{aligned} \epsilon(\mathbf{x},t) &= c\rho(\mathbf{x})h(\mathbf{x},t) \\ h(\mathbf{x},t) &= \frac{d}{dt}T(\mathbf{x},t) \end{aligned}$$

[00169] Combining all these elements, we obtain:

$$c\rho(\mathbf{x},t)\frac{d}{dt}T(\mathbf{x},t) = \nabla \cdot (\lambda \nabla T(\mathbf{x},t)) + \sigma(\mathbf{x},t) \tag{28}$$

Discrete Representation of Images

[00170] Some algebraic tools used to model images will now be recalled from the above-description. An image is composed of two distinctive parts: the image support (pixels) and some field quantity associated with each pixel. This quantity may be scalar (e.g. gray level), vectorial (e.g. color, multispectral, optical flow) or tensorial (e.g. Hessian). We model the image support in terms of cubical complexes, chains and boundaries. With these concepts, we are able to give a formal description of an image support of any dimension. For quantities, we introduce the concept of cochains, which are representations of fields over a cubical complex. For the use of these concepts in image processing, see [16].

[00171] As discussed hereinabove, an image is a complex of unit cubes usually called pixels. A pixel $\gamma \subset \Re^n$ is a product

$$\gamma = I_1 \times I_2 \times \ldots \times I_n$$

where I_j is either a singleton or an interval of unit length with integer endpoints. Thus I_j is either the singleton {k} and is said to be a degenerate interval, or the closed interval [k, k+1] for some $k \in Z$. The number q $\in$ {0,1,...,n} of non-degenerate intervals is by definition the dimension of γ, which is called a q-pixel. Figure 6 illustrates three elementary pixels in $\Re^2$. For $q \geq 1$, let $J = \{k_0, k_1, .. k_{q-1}\}$ be the ordered subset {1,2, ...,n} of indices for which $I_{j_k} = [a_j, b_j]$ is non-degenerate. Let us define

$$A_{k_j}\sigma \;=\; I_1 \times \ldots I_{k_j-1} \times \{a_j\} \times I_{k_j+1} \times \ldots \times I_n$$

and

$$B_{k_j}\sigma \;=\; I_1 \times \ldots I_{k_j-1} \times \{b_j\} \times I_{k_j+1} \times \ldots \times I_n$$

[00172] The A_{k_j} and the B_{k_j} are called the (q–1)-faces of σ. One can define the (q–2)-faces, ..., down to the 0-faces of σ in the same way. The faces of γ different from γ itself are called its proper faces.

[00173] By definition, a natural orientation of the cube is assumed for each pixel. Suppose that γ denotes a particular positively oriented q-pixel. It is natural to denote the same pixel with opposite orientation by $-\gamma$. Examples of orientations are given in Figure 6. A cubical complex in $\Re^n$ is a finite collection K of q-pixels such that every face of any pixel of the image support is also a pixel in K and the intersection of any two pixels of K is either empty or a face of each of them. For example, traditional 2D image models only considered pixels as 2D square elements. The definitions presented above allow us to consider 2-pixels (square elements), 1-pixels (line elements) and 0-pixels (point elements) simultaneously.

[00174] In order to write the image support in algebraic form, we introduce the concept of chains. Any set of oriented q-pixels of a cubical complex can be written in algebraic form by attributing to them the coefficient 0,1 or –1, if they are not in the set or if they should or should not be taken with positive orientation, respectively. In order to represent weighted domains, we must allow arbitrary integer multiplicity for each q-pixel.

[00175] Given a topological space $X \subset \Re^n$ in terms of a cubical complex, we get a free abelian group $C_q(X)$ generated by all the q-pixels. The elements of this group are called q-chains and they are formal linear combinations of q-pixels [16]. A formal expression for a q-chain c_q is $c_q = \sum_{\gamma_i \in K} \lambda_i \gamma_i$ where $\lambda_i \in Z$.

[00176] The last step needed for the description of the image plane is the introduction of the concept of a boundary of a chain. Given a q-pixel γ, we

define its boundary $\partial\gamma$ as the (q-1)-chain corresponding to the alternating sum of its (q-1)-faces. The sum is taken according to the orientation of the (q-1)-faces with respect to the orientation of the q-pixel. We say that a (q-1)-face of γ is positively oriented relative to the orientation of γ if its orientation is compatible with the orientation of γ. By linearity, the extension of the definition of boundary to arbitrary q-chains is easy. For instance, in Figures 6(b) and 6(c), the boundary of the 1-pixel a is $x_2 - x_1$ and the boundary of the 2-pixel A is $a+b-c-d$; we say that a and b are positively oriented with respect to the orientation of A but c and d are negatively oriented with respect to the orientation of A. Let us notice that the boundary of a 1-pixel is always the difference between its boundary points. The boundary can be defined recursively. Given a (q-1)-chain and a q-chain γ_q defined as $\gamma_q = \gamma_{q-1} \times [a,b]$, the boundary of γ_q can be recursively written as

$$\partial\gamma_q = \partial\gamma_{q-1} \times [a,b] + (-1)^{(q-1)}(\gamma_{q-1} \times \{b\} - \gamma_{q-1} \times \{a\}) \quad (29)$$

[00177] In order to model the pixel quantity over the image plane, we look for an application F which associates a global quantity with all q-pixels γ of a cubical complex. We denote this by $<F,\gamma>$. This quantity may be any mathematical entity such as a scalar, a vector, etc. For two adjacent q-pixels γ_1 and γ_2, F must satisfy $<F,\lambda_1\gamma_1+\lambda_2\gamma_2>=\lambda_1<F,\gamma_1>+\lambda_2<F,\gamma_2>$, which means that the sum of the quantity over each pixel is equal to the quantity over the two pixels. The resulting transformation $F:C_q(X)\rightarrow\Re$ is called a q-cochain and is used as a representation of a quantity over the cubical complex.

[00178] Finally we need an operator which associates a global quantity with the (q+1)-pixels according to the global quantities given on their q-faces. Given a q-cochain F, we define an operator ∂, called the coboundary operator, which transforms F into a (q+1)-cochain ∂F such that

$$< \delta\mathcal{F}, \gamma > = < \mathcal{F}, \partial\gamma > \tag{30}$$

for all (q+1)-chains γ. The coboundary is defined as the signed sum of the physical quantities associated with the q-faces of γ. The sum is taken according to the relative orientation of the q-faces of the (q+1)-pixels of γ with respect to the orientation of the pixels. Figure 7 presents an example of the coboundary operation for a 2-pixel.

[00179] With this image model in hand, we will use the basic laws described hereinabove to rewrite the global heat transfer equation in algebraic terms [43].

Representation of the Heat Transfer Equation

[00180] The process for representing the heat transfer equation in terms of algebraic topology can be summarized as follows. The image support is subdivided into cubical complexes. Basic laws are applied to pixels of various dimensions. These laws involve the computation of global quantities on pixels, expressed as cochains. Some of these laws link global quantities on pixels with global quantities on their boundaries and hence are expressed as coboundaries. The other laws are expressed as linear transformations between pairs of cochains. The topological formalism of cochain and coboundary is a generic one; that is, it does not offer computational rules. The cochains must be instantiated depending on the problem to be considered.

[00181] The basic laws presented hereinabove will be reformulated in a topological way and then to give computational rules for cochains in the context of the heat transfer problem. Since we want to represent two kinds of global values over the spatio-temporal image, we use two complexes. The first complex is associated with global values corresponding to projections onto the tangential part of the domain (e.g. global thermal tension) while the second complex refers to values related to projections onto the normal part of the

domain (e.g. heat entering a particle). These two distinct orientations (see Figures 8(a) and 8(b)) give rise to two different complexes.

Global Heat Transfer

[00182] Let us assume that we have an image with n spatial dimensions and r pixels. Suppose also that we have a time interval [t_0, t_l] which we can split into l equal sub-intervals [t_k, t_{k+1}], k ∈ [0, l–1]. Let us consider an n-complex representing the subdivided spatial support of the image $K^{s'}$. We can consider an (n+1)-complex representing the spatio-temporal support of the image:

$$\mathcal{K}^s = \mathcal{K}^{s'} \times [t_k, t_{k+1}], \forall k \in [0, l-1]$$

[00183] Now, let us consider γ_E, an (n+1)-pixel of K^s, and the following cochain, defined as $<\varepsilon, \gamma_E>$. Thus, we therefore need to define which value to use as cochain ε in the heat transfer problem. Let us define ε as the global energy variation on the (n+1)-pixel γ_E:

$$< \mathcal{E}, \gamma_E >= \int_{\gamma_E} \epsilon(\mathbf{x}, t)\, d\gamma_E \qquad (31)$$

where $d\gamma_E$ is an infinitesimal part of the domain represented by γ_E. Now, using the global version of Equation 18, we obtain :

$$\int_{\gamma_E} \epsilon(\mathbf{x}, t)\, d\gamma_E = \int_{\gamma_E} \alpha(\mathbf{x}, t)\, d\gamma_E + \int_{\gamma_E} \sigma(\mathbf{x}, t)\, d\gamma_E$$

[00184] From this equation, we define two more cochains, representing first, the global diffusion:

$$< \mathcal{D}, \gamma_E >= \int_{\gamma_E} \alpha(\mathbf{x}, t)\, d\gamma_E \tag{32}$$

and second, the global source:

$$< \mathcal{S}, \gamma_E >= \int_{\gamma_E} \sigma(\mathbf{x}, t)\, d\gamma_E$$

[00185] Thus, we have the following relation between the three cochains:

$$< \mathcal{E}, \gamma_E >=< \mathcal{D}, \gamma_E > + < \mathcal{S}, \gamma_E > \tag{33}$$

[00186] The rules used for cochains ε and **D** are then decomposed into basic laws. The rule for cochain **S** is not decomposed since we assume that its global value is known on γ_E. Let us finally mention that both steady and unsteady heat transfer problems can be considered using Equation 33 by setting respectively, cochains ε and **S**, to zero.

Global Temperature Variation

[00187] Let us consider another n-complex, $K^{p'}$, representing the subdivided spatial domain of the image. We can define an (n+1)-complex representing the spatio-temporal image:

$$\mathcal{K}^p = \mathcal{K}^{p'} \times [t_k, t_{k+1}], \forall k \in [0, l-1]$$

[00188] Let us consider γ_H, a 1-pixel of K^p defined as $\mathbf{x}_i \times [t_k, t_{k+1}]$, $i \in [1, r]$, $k \in [0, l-1]$ where $\mathbf{x}_i$ is a 0-pixel of $K^{p'}$. Let us also consider a 0-cochain T and a 1-cochain H such that

$$< \mathcal{H}, \gamma_H > = < \delta\mathcal{T}, \gamma_H > = < \mathcal{T}, \partial\gamma_H > \quad (34)$$

[00189] Figure 9 presents examples of T and H for K^p of dimension 3.

[00190] Applying Equation 29, we find that the boundary of γ_H is $\partial\gamma_H$ = $x_i \times \{t_{k+1}\}$ – $x_i \times \{t_k\}$. According to the linearity of the cochain, the computational rule relating the global value associated to γ_H with the values at its boundary $x_i \times \{t_k\}$ and $x_i \times \{t_{k+1}\}$ is

$$< \mathcal{T}, \partial\gamma_H > = < \mathcal{T}, \mathbf{x}_i \times \{t_{k+1}\} - \mathbf{x}_i \times \{t_k\} > = < \mathcal{T}, \mathbf{x}_i \times \{t_{k+1}\} > - < \mathcal{T}, \mathbf{x}_i \times \{t_k\} > \quad (35)$$

[00191] This equation is general and applies to many problems. To define which values to use as 0-cochain and 1-cochain, let us take the global version of Equation 27 on γ_H and apply the fundamental calculus theorem:

$$\int_{\gamma_H} h(\mathbf{x}, t)\, d\gamma_H = \int_{t_k}^{t_{k+1}} \frac{d}{dt} T(\mathbf{x}_i, t)\, dt = T(\mathbf{x}_i, t_{k+1}) - T(\mathbf{x}_i, t_k)$$

[00192] Looking at this equation, we see that it is similar to Equation 35. Thus we define T = T(x, t). The location of the unknown temperatures we want to compute must correspond to the 0-pixels of K^p. In order to fulfill Equation 34, we must therefore have

$$< \mathcal{H}, \gamma_H > = \int_{\gamma_H} h(\mathbf{x}, t) d\gamma_H \quad (36)$$

which is called the global temperature variation. These three equations are extended by linearity to a 1-chain of K^p defined as $\gamma \times [t_k, t_{k+1}]$, where γ is an arbitrary 0-chain of $K^{p'}$.

Global Energy Variation

[00193] We want to link cochains H and ε, representing the global temperature variation and the global energy variation, respectively. For this purpose, we need to represent Equation 26. The two cochains are not from the same cubical complex (H is from K^p and ε is from K^s), and moreover, Equation 26 is material-dependent; therefore they cannot be linked exactly. However, we can express this link as a linear transformation

$$H \xrightarrow{\Gamma} \varepsilon$$

[00194] Recalling Equation 31, we have

$$< \mathcal{E}, \gamma_E >= \int_{\gamma_E} \epsilon(\mathbf{x}, t) d\gamma_E = \int_{\gamma_E} c\rho(\mathbf{x}) h(\mathbf{x}, t)\, d\gamma_E \tag{37}$$

[00195] Unfortunately, we do not know the value of ρ(**x**) or h(**x**, t) at all points of the volume. Consequently we must approximate these two fields over the volume. We denote the approximations by $\tilde{\rho}$(**x**) and $\tilde{h}$(**x**, t). For one 1-pixel γ_H, defined as $x_i \times [t_k, t_{k+1}]$, the approximation is performed piecewise such that $\tilde{h}$(**x**, t) must fulfill Equation 36.

$$\int_{t_k}^{t_{k+1}} \tilde{h}(\mathbf{x}_i, t) =< \mathcal{H}, \gamma_H > \tag{38}$$

[00196] Equation 37 thus becomes

$$< \mathcal{E}, \gamma_E >= \int_{\gamma_E} c\tilde{\rho}(\mathbf{x})\tilde{h}(\mathbf{x}, t)\, d\gamma_E = f_e(c, \mathcal{H}) = \Gamma \tag{39}$$

where dV is an infinitesimal part of γ_E which depends on the choice of the approximation functions $\tilde{\rho}$(**x**) and $\tilde{h}$(**x**, t) and the position of K^s with respect to K^p.

Global Diffusion

[00197] Let us consider an n-cochain Q and an (n+1)-cochain D defined by the coboundary

$$<\mathcal{D},\gamma_E>=<\delta\mathcal{Q},\gamma_E>=<\mathcal{Q},\partial\gamma_E> \tag{40}$$

[00198] Figure 10 presents examples of Q and D for K' of dimension 3. Let us assume that the n-faces γ_{Q_i} of γ_E are positively oriented relative to γ_E. According to the linearity of the cochain, the computational rule is

$$<\mathcal{D},\gamma_E>=\sum_{\gamma_{Q_i}\in\partial\gamma_E}<\mathcal{Q},\gamma_{Q_i}> \tag{41}$$

[00199] Again, this equation is general; hence we need to find a global value for the (n+1)-cochain D, which can be computed by summing the global values at the boundary of γ_E. According to Equation 32, we have

$$<\mathcal{D},\gamma_E>=\int_{\gamma_E}\alpha(\mathbf{x},t)\,d\gamma_E$$

[00200] Applying the divergence theorem to this equation, we obtain

$$g_t(\mathbf{x})=\frac{1}{4\pi t}e^{-\left(\frac{x^2+y^2}{4t}\right)}$$

where **n(x, t)** is the normal vector to an infinitesimal part of the domain represented by γ_Q. This last equation is in the form of a coboundary (Equation 41), from which we define

$$<\mathcal{Q},\gamma_{Q_i}>=\int_{\gamma_{Q_i}}-\mathbf{q}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x},t)\,d\gamma_{Q_i} \tag{42}$$

[00201] Again, the previous definitions can be extended by linearity to arbitrary (n+1)-chains of K^s. We recall that there is absolutely no approximation in these equations.

Global Thermal Tension

[00202] Let us consider a 1-pixel γ_G of K^p defined as $\gamma \times t_k$, $k \in [0, l-1]$, where γ is a 1-pixel of $K^{p'}$ whose boundary is defined as $\partial\gamma = \mathbf{x}_j - \mathbf{x}_i$, $i, j \in [1, r]$. Let us also consider a 0-cochain T and a 1-cochain G defined by the coboundary

$$< \mathcal{G}, \gamma_G >=< \delta\mathcal{T}, \gamma_G >=< \mathcal{T}, \partial\gamma_G > \tag{43}$$

[00203] Figure 11 presents examples of T and G for one 3-pixel of K^p. The boundary of γ_G is $\partial\gamma_G = \mathbf{x}_j \times \{t_k\} - \mathbf{x}_i \times \{t_k\}$. According to the linearity of the cochain, the computational rule relating the global value associated with γ_G to the values at $\mathbf{x}_i \times \{t_k\}$ and $\mathbf{x}_j \times \{t_k\}$ is

$$< \mathcal{T}, \partial\gamma_G >=< \mathcal{T}, \mathbf{x}_j \times \{t_k\} - \mathbf{x}_i \times \{t_k\} >=< \mathcal{T}, \mathbf{x}_j \times \{t_k\} > - < \mathcal{T}, \mathbf{x}_i \times \{t_k\} > \tag{44}$$

[00204] To define which values to use as cochains G and T let us take the global form of Equation 24 on γ_G

$$\int_{\gamma_G} \mathbf{g}(\mathbf{x}, t) \cdot d\gamma_G = \int_{\mathbf{x}_i}^{\mathbf{x}_j} \mathbf{g}(\mathbf{x}, t_k) \cdot d\gamma = \int_{\mathbf{x}_i}^{\mathbf{x}_j} -\nabla T(\mathbf{x}, t_k) \cdot d\gamma \tag{45}$$

where $d\gamma$ is an infinitesimal part of γ. Since g(x, t) is a spatial conservative field, we can apply the line integral theorem [26, 41] saying that for a conservative field F(x) = ∇f(x) and two points A and B, in an open connected region containing **F(x)**, the integral of the tangential part of **F(x)** along the curve R joining A and B is independent of the path (Figure 12):

$$\int_A^B \mathbf{F}(\mathbf{x}) \cdot dR_1 = \int_A^B \mathbf{F}(\mathbf{x}) \cdot dR_2 = \int_A^B \mathbf{F}(\mathbf{x}) \cdot dR_3 = f(B) - f(A)$$

[00205] From this theorem, we can rewrite Equation 45 as

$$\int_{\mathbf{x}_i}^{\mathbf{x}_j} \mathbf{g}(\mathbf{x}, t_k) \cdot d\gamma = (-T(\mathbf{x}_j, t_k)) - (-T(\mathbf{x}_i, t_k)) = T(\mathbf{x}_i, t_k) - T(\mathbf{x}_j, t_k) \tag{46}$$

[00206] Looking at Equation 46, we see that it is similar to Equation 44. Thus we define T = T(**x**, t). Consequently, the location of the unknown temperatures we want to compute must correspond to the 0-pixels of K^p which is coherent with the conclusions hereinabove. In order to fulfill Equation 43, we must have

$$< \mathcal{G}, \gamma_G >= \int_{\gamma_G} -\mathbf{g}(\mathbf{x}, t) \cdot d\gamma_G \tag{47}$$

[00207] The previous definitions are extended by linearity to 1-chains of K^p defined as $\gamma \times \{t_k\}$, where γ is an arbitrary 1-chain of $K^{p'}$.

Heat Flow Density

[00208] The coboundaries $<Q, \partial\gamma_E>$ (Equation 40) and $<T, \partial\gamma_G>$ (Equation 43) provide exact global versions of Equation 23 on K' and Equation 24 on K^p, respectively. In order to complete the diffusion term, we need to represent Equation 25, which links local values **g(x,t)** and **q(x,t)**. Equation 25 is a constitutive equation and cannot be represented by a topological equation. However, we can find a relation transforming cochain G into cochain Q:

$$< \mathcal{G}, \gamma_G > \overset{\wedge}{\leftrightarrow} < \mathcal{Q}, \gamma_Q >$$

as a global counterpart for Equation 25. To find this transformation, we recall Equation 42:

$$< \mathcal{Q}, \gamma_{Q_i} >= \int_{\gamma_{Q_i}} -\mathbf{q}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x},t)\,d\gamma_{Q_i} = \int_{\gamma_{Q_i}} -\lambda\mathbf{g}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x},t)\,d\gamma_{Q_i}$$

relating cochain Q to field **g(x,t)**. Unfortunately, field **g(x,t)** is not known, so we have to approximate it with a field $\tilde{g}$ (x,t). Let us consider γ_n, an n-pixel of K^p defined as $\gamma_n = \gamma_x \times \{t_k\}$, $k \in [0, l-1]$ where γ_x is an n-pixel of $K^{p'}$. This approximation is performed piecewise such that for each 1-face γ_G of γ_n, $\tilde{g}$ (x,t) satisfies

$$\int_{\gamma_G} -\tilde{\mathbf{g}}(\mathbf{x},t)\cdot dR =< \mathcal{G}, \gamma_G > \tag{48}$$

where dR is an infinitesimal part of the domain represented by γ_G. Equation 25 is then applied to obtain $\tilde{q}$ (x,t):

$$\tilde{\mathbf{q}}(\mathbf{x},t) = \lambda\tilde{\mathbf{g}}(\mathbf{x},t)$$

at all points of the domain. Equation 42 becomes

$$< \mathcal{Q}, \gamma_{Q_i} >= \int_{\gamma_{Q_i}} -\tilde{\mathbf{q}}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x},t)\,d\gamma_{Q_i} = f_g(\lambda, \mathcal{G}) \tag{49}$$

[00209] The transformation we are looking for is thus

$$\Lambda = f_g(\lambda, \mathcal{G})$$

which depends on the choice of an approximation function $\tilde{g}$ (x,t) and the position of K' with respect to K^p.

Boundary Conditions

[00210] The decomposition process we have presented herein is carried out with the assumption that all the needed quantities surrounding a pixel are known. For instance, in the steady state heat transfer problem, for a particular (n+1)-pixel, the cochain S must be known for all other surrounding (n+1)-pixels, that is, there are as many equations as variables.

[00211] Unfortunately, this assumption is not verified at the borders of the image. Thus, as in solving the PDE, certain boundary conditions must be imposed to specify the gray-level conditions at the boundary of the image. For instance, these conditions may prescribe the values of either cochain T (Dirichlet boundary conditions) or cochain Q (Neumann boundary conditions).

Summary of the Algorithm

[00212] We will now summarize the algorithm used to find an expression of the temperatures at time t_{k+1} as a function of the temperatures at time t_k. The input data for this algorithm are the cochain S and the Dirichlet boundary conditions. That is, T is known for all pixels on the boundary of the image, which includes the values at time t_0.

1. Choose the positions for $K^{p'}$ and $K^{s'}$.

2. Compute ε as a function of H:

   (a) Choose the approximation functions $\tilde{h}$ (x,t) and $\tilde{p}$ (x).

   (b) Apply Equation 38 to find $\tilde{h}$ (x,t_k, t_l) as a function of H.

   (c) Apply Equation 39 to find the transformation Γ, expressing ε as a function of H.

3. Apply Γ and Equation 34 to find ε as a function of T.

4. Compute Q as a function of G:

   (a) Choose the approximation function $\tilde{g}$ (x,t).

   (b) Apply Equation 48 to find $\tilde{g}$ (x,t) as a function of G.

   (c) Apply Equation 49 to find the transformation Λ, expressing Q as a function of G.

5. Apply Equation 40, Λ and Equation 43 to find D as a function of T.

6. Apply Equation 33 to obtain an equation of the temperatures at time t_{k+1} as a function of the temperatures at time t_k

[00213] Figure 13 presents an overview of the computational scheme.

Applications

Linear Isotropic Diffusion

[00214] One of the most direct applications of the heat transfer equation is the isotropic diffusion of gray-level intensities; that is, smoothing. For a 2D image I(**x**), with **x** = (x,y), the resolution of the PDE

$$\frac{\partial}{\partial t}I(\mathbf{x},t) = \nabla^2 I(\mathbf{x},t) \tag{50}$$

is equivalent to the convolution

$$I(\mathbf{x},t) = (I * g_t)(\mathbf{x})$$

where

$$g_t(\mathbf{x}) = \frac{1}{4\pi t}e^{-\left(\frac{x^2+y^2}{4t}\right)}$$

is a Gaussian with variance $\sigma^2 = 2t$ [25]. One can see t as the scale of the smoothing operation. Let us assume that the Laplacian image at scale t

$$L(\mathbf{x}, t) = \nabla^2 I(\mathbf{x}, t) \tag{51}$$

is known. One can consider this equation as a steady state heat transfer problem with T(x,t) = I(x,t), σ(x,t) = –L(x,t) and λ = 1.

[00215] We want to solve Equation 51 for local I(x,t) located at the center of each image pixel. Employing the process presented hereinabove, we first position the two cubical complexes representing two subdivisions of the image plane. As stated hereinabove, the primary complex $K^{p'}$ is defined with 0-pixels corresponding to pixel centers. For the sake of simplicity, $K^{s'}$ corresponds to the image pixels; that is, the secondary 2-pixels γ_s are rectangular and symmetrically staggered relative to the 1-pixels of K^p and the 1-pixels γ_q of K^s intersect orthogonally in the centers of the primary 1-pixels. Since there is no variation in steady-state heat transfer over time, we drop the time parameter. This means that $K^p = K^{p'}$, $K^s = K^{s'}$ and the time integral in cochain computation are dropped. We saw that the approximation function f$_g$ depends on the position of K^s with respect to K^p.

[00216] Figure 14 shows the two complexes for a 5 × 5 image. Positioning the 1-faces of K^s such as each passes through the center point between two 0-pixels of K^p allows us to compute a polynomial function of order 1 with the same accuracy as that obtained using one of order 2 [37, 34].

[00217] We need a global value for the 2-cochain $<S,\gamma_E>$. If we assume that a pixel value represents the global value of intensity, we can

directly set $<S, \gamma_E> = -L(\mathbf{x})$. This assumption is reasonable if we look at image acquisition as a process which accumulates the total number of photons within a global area corresponding to the pixel [22].

[00218] We must choose an approximation function $\tilde{g}(\mathbf{x})$. For simplicity, we assume that $\tilde{g}(\mathbf{x})$ arises from a bilinear approximation; that is:

$$\tilde{\mathbf{g}}(\mathbf{x}) = (a + by) \cdot \vec{i} + (c + dx) \cdot \vec{j}$$

[00219] Given a 2-pixel γ_p of K^p, $\tilde{g}(\mathbf{x})$ must satisfy Equation 48 for each 1-face of γ_p. As an example, let us find the coefficients a, b, c and d for such a pixel defined as in Figure 15. We have

$$\begin{aligned} \mathcal{G}_1 &= \int_0^\Delta -\tilde{\mathbf{g}}(x, 0) \cdot \vec{i} dx \\ \mathcal{G}_2 &= \int_0^\Delta -\tilde{\mathbf{g}}(\Delta, y) \cdot \vec{j} dy \\ \mathcal{G}_3 &= \int_0^\Delta -\tilde{\mathbf{g}}(x, \Delta) \cdot \vec{i} dx \\ \mathcal{G}_4 &= \int_0^\Delta -\tilde{\mathbf{g}}(0, y) \cdot \vec{j} dy \end{aligned}$$

from which we obtain

$$\tilde{\mathbf{g}}(\mathbf{x}) = -\frac{1}{\Delta}\left[\left(\mathcal{G}_1 + \frac{(\mathcal{G}_3 - \mathcal{G}_1)}{\Delta} y\right) \cdot \vec{i} + \left(\mathcal{G}_4 + \frac{(\mathcal{G}_2 - \mathcal{G}_4)}{\Delta} x\right) \cdot \vec{j}\right], \mathbf{x} \in \gamma_p \quad (52)$$

$\tilde{g}(\mathbf{x})$ is thus a piecewise function of G, but as G is computed from T, we can also express $\tilde{g}(\mathbf{x})$ as a function of T. For each primary 2-pixel, we apply Equation 25 to obtain $\tilde{q}(\mathbf{x}) = \tilde{g}(\mathbf{x})$.

[00220] The next step is to compute $<Q,\gamma_Q>$ for K^s from Equation 49. Each secondary 2-pixel γ_s intersects with four primary 2-pixels, γ_{pa}, γ_{pb}, γ_{pc} and γ_{pd}. There are four segments in the approximation function $\tilde{q}$(x) corresponding to the four primary 2-pixels; that is, $\tilde{q}_a$(x), $\tilde{q}_b$(x), $\tilde{q}_c$(x) and $\tilde{q}_d$(x). Figure 16 illustrates γ_E. We find cochain $<Q,\gamma_Q>$ corresponding to the four 1-faces of γ_E:

$$\begin{aligned}
\mathcal{Q}_1 &= \int_0^{\Delta/2} -\tilde{\mathbf{q}}_a(\Delta/2,y)\cdot\vec{i}dy + \int_{-\Delta/2}^{0} -\tilde{\mathbf{q}}_b(\Delta/2,y)\cdot\vec{i}dy \\
&= \frac{3\mathcal{G}_1}{4}+\frac{\mathcal{G}_3}{8}+\frac{\mathcal{G}_5}{8} \\
\mathcal{Q}_2 &= \int_0^{\Delta/2} -\tilde{\mathbf{q}}_b(x,-\Delta/2)\cdot(-\vec{j})dx + \int_{-\Delta/2}^{0} -\tilde{\mathbf{q}}_c(x,-\Delta/2)\cdot(-\vec{j})dx \\
&= -\frac{3\mathcal{G}_7}{4}-\frac{\mathcal{G}_6}{8}-\frac{\mathcal{G}_9}{8} \\
\mathcal{Q}_3 &= \int_0^{\Delta/2} -\tilde{\mathbf{q}}_a(x,\Delta/2)\cdot\vec{j}dx + \int_{-\Delta/2}^{0} -\tilde{\mathbf{q}}_d(x,\Delta/2)\cdot\vec{j}dx \\
&= \frac{3\mathcal{G}_4}{4}+\frac{\mathcal{G}_2}{8}+\frac{\mathcal{G}_{11}}{8} \\
\mathcal{Q}_4 &= \int_0^{\Delta/2} -\tilde{\mathbf{q}}_d(-\Delta/2,y)\cdot(-\vec{i})dy + \int_{-\Delta/2}^{0} -\tilde{\mathbf{q}}_c(-\Delta/2,y)\cdot(-\vec{i})dy \\
&= -\frac{3\mathcal{G}_{10}}{4}-\frac{\mathcal{G}_{12}}{8}-\frac{\mathcal{G}_8}{8}
\end{aligned} \qquad (53)$$

[00221] Using Equation 41, we have

$$< \mathcal{D}, \gamma_E >= \mathcal{Q}_1 + \mathcal{Q}_2 + \mathcal{Q}_3 + \mathcal{Q}_4 \qquad (54)$$

[00222] Substituting Equation 43 in Equation 53, Equation 53 in Equation 54 and Equation 54 in Equation 33, we can now express $< S, \gamma_E >$ as a function of T . As an example, we present $< S, \gamma_E >$ for a 2-pixel γ_E, defined as in Figure 16:

$$< \mathcal{S}, \gamma_s >= -3\mathcal{T}_{0,0} + \frac{1}{2}[\mathcal{T}_{0,1} + \mathcal{T}_{1,0} + \mathcal{T}_{0,-1} + \mathcal{T}_{-1,0}] + \frac{1}{4}[\mathcal{T}_{-1,1} + \mathcal{T}_{1,1} + \mathcal{T}_{1,-1} + \mathcal{T}_{-1,-1}] \qquad (55)$$

[00223] For each non-border pixel (represented by a secondary 2-pixel), we get an equation in the form of Equation 55. For the border pixels, we set T = I(**x**). Solving this system, we obtain the smoothed image I(**x**,t) = T.

Optical Flow

[00224] An indirect application of the heat transfer equation is the computation of optical flow for a 2D image sequence I(**x**,t), using the Horn and Schunk [29] algorithm. It can be shown that the velocity vector **u**(**x**,t) = (u(**x**,t), v(**x**,t)) satisfies the following constraint arising from variational calculus (for greater legibility, we have dropped the (**x**,t)) :

$$\begin{aligned} I_x^2 u + I_x I_y v &= \alpha^2 \nabla^2 u - I_x I_t \\ I_x I_y u + I_y^2 v &= \alpha^2 \nabla^2 v - I_y I_t \end{aligned} \qquad (56)$$

where α is a weighting factor and I_x, I_y and I_t are the first derivatives of I(**x**,t) in x, y and t, respectively. Let us rewrite Equation 56 in the following vectorial form:

$$\nabla I(\nabla I \cdot \mathbf{u}) = \alpha^2 \nabla^2 \mathbf{u} - I_t \nabla I$$

where we define $\nabla^2 \mathbf{u} = (\nabla^2 u,\ \nabla^2 v)$. Reorganizing the terms of Equation 56, we get the following equation:

$$\alpha^2 \nabla^2 \mathbf{u} = \nabla I (\nabla I \cdot \mathbf{u}) + I_t \nabla I. \qquad (57)$$

[00225] Taking σ(**x**, t) = $-\nabla$I(∇I . **u**) – I_t ∇I as a heat source, Equation 57 can be seen as a steady state heat transfer equation in which the cochain T corresponds to **u**(**x**, t) and $\lambda = \alpha^2$. It can thus be decomposed using the method descrbed hereinabove. The cochain T is **u**(**x**, t), and we get the relation:

$$-\frac{I_t \nabla I}{\alpha^2} = -3\mathcal{U}_{0,0} + \nabla I(\nabla I \cdot \mathcal{U}_{0,0}) + \frac{1}{2}[\mathcal{U}_{0,1} + \mathcal{U}_{1,0} + \mathcal{U}_{0,-1} + \mathcal{U}_{-1,0}] + \frac{1}{4}[\mathcal{U}_{-1,1} + \mathcal{U}_{1,1} + \mathcal{U}_{1,-1} + \mathcal{U}_{-1,-1}]$$

[00226] For the same reasons as in the linear diffusion problem, special considerations are needed at the borders of the image. We assume zero velocity at the borders of the image and solve the system to get the velocity field for each point of the image.

Nonlinear Diffusion

[00227] Linear isotropic diffusion reduces noise but also blurs edges. As the scale increases, edges tend to be harder to identify [43]. One possible way of reducing this effect might be to consider the heat conduction coefficient λ as a field function dependent on the magnitude of the edges; that is:

$$\frac{\partial}{\partial t} I(\mathbf{x}, t) = \nabla \cdot (g(|\nabla I(\mathbf{x}, t)|^2) \nabla I(\mathbf{x}, t)) \qquad (58)$$

which corresponds to Equation 28 with λ(**x**,t) = g(|∇I(**x**,t)|^2), T(**x**,t) = I(**x**,t), ρ(**x**) = 1, c = 1 and σ(**x**,t) = 0 (i.e., unsteady transfer with no source). The conduction function g(s) must display the following behavior: in constant regions, there should be linear isotropic diffusion (Equation 50), that is g(|∇I(**x**,t)|^2) = 1 for |∇I(**x**,t)|^2 = 0, and almost no diffusion when the magnitude of the edge is great; that is, $g(|\nabla I(x,t)|^2) = 0$ for $|\nabla I(x,t)|^2 \to \infty$. Perona and Malik [38] proposed the following functions:

$$g(s) = \frac{1}{1+\frac{s^2}{k^2}}, \qquad (k > 0)$$

and

$$g(s) = e^{-\frac{s^2}{k^2}}, \qquad (k > 0)$$

[00228] The parameter k in these functions is difficult to set because it controls the threshold of diffusion but also the steepness of the function [35]. We prefer to use the function

$$g(s) = \frac{1}{2}\tanh\left(\gamma(k-s)+1\right)$$

where k and γ control the threshold and the steepness, respectively. We then solve Equation 58 for a particular t (the scale) with initial conditions

$$I(\mathbf{x},0) = I(\mathbf{x})$$

where I(x) is the original image.

[00229] Let us assume that we have l time steps $\Delta t = t/l$. First, we use the same cubical complexes $K^{p'}$ and $K^{s'}$ as hereinabove and we define

$$\begin{aligned} \mathcal{K}^p &= \mathcal{K}^{p'} \times [t_k, t_{k+1}] \\ \mathcal{K}^s &= \mathcal{K}^{s'} \times [t_k, t_{k+1}], \quad t_k = k\Delta t, \quad \forall k \in [0, l-1] \end{aligned}$$

[00230] Secondly, we make an assumption about the spatial behavior of h(x, t); that is, we choose the approximation function $\tilde{h}$ (x, t). For a 3-pixel γ_E

as defined in Figure 17, let us assume that H is the mean value over $[-\Delta/2,\Delta/2] \times [-\Delta/2,\Delta/2]$. We thus have $\varepsilon = H$; that is, using Equation 34, we can also write

$$< \mathcal{E}, \gamma_E >= \mathcal{T}^1 - \mathcal{T}^0 \quad (59)$$

[00231] For the sake of simplicity, we use the same spatial bilinear approximation function $\tilde{g}$ (x, t) as hereinabove. We have to approximate the behavior over a time step. Some common assumptions about time variation may be generalized by proposing [37]:

$$\int_{t_k}^{t_{k+1}} A(t)\,dt = (wA(t_{k+1}) + (1-w)A(t_k))\,\Delta t, \quad 0 \leq w \leq 1$$

where A(t) is some quantity and w is a weighting factor [37]. Some values of w lead to well-known schemes: 1) w leads to the explicit scheme; that is, the value at t_k prevails for the entire time interval except at time t_{k+1}. 2) w = 1 leads to the fully implicit scheme; that is, the value changes at time t_k from $A(t_k)$ to $A(t_{k+1})$ and stays there throughout the whole time interval. 3) w = 0.5 leads to the semi-implicit or Crank-Nicolson scheme; that is, there is a linear variation of A(t) . We propose to use the implicit scheme because for large values of Δt, it best emulates long term time behavior for heat [37]. That is for a 3-pixel, we have for w = 1

$$\tilde{\mathbf{g}}(\mathbf{x}) = -\frac{1}{\Delta}\left[\left(\mathcal{G}_1^1 + \frac{(\mathcal{G}_3^1 - \mathcal{G}_1^1)}{\Delta}y\right)\cdot\vec{i} + \left(\mathcal{G}_4^1 + \frac{(\mathcal{G}_2^1 - \mathcal{G}_4^1)}{\Delta}x\right)\cdot\vec{j}\right]\Delta t, \mathbf{x} \in \gamma_p, t \in [0, \Delta t]$$

[00232] In order to obtain the local function $\tilde{q}$ (x, t) we apply Equation 25:

$$\tilde{\mathbf{q}}(\mathbf{x}, t) = \lambda(\mathbf{x}, t)\tilde{\mathbf{g}}(\mathbf{x}, t)$$

where $\lambda(\mathbf{x}, t) = g(|\nabla I(\mathbf{x},t)|^2)$. As $\nabla I(\mathbf{x},t)$ is a spatially sampled image where samples are located at the 0-pixels of K^p, we must approximate the local values of $\lambda(\mathbf{x}, t)$. For the sake of simplicity, we once again use a bilinear approximation; that is:

$$\tilde{\lambda}(\mathbf{x}) = a + bx + cy + dxy$$

[00233] For a 2-pixel of K^p, as illustrated in Figure 18, we obtain

$$\tilde{\lambda}(\mathbf{x}, t) = \lambda_{0,0} + \frac{1}{\Delta}((\lambda_{1,0} - \lambda_{0,0})x + (\lambda_{0,1} - \lambda_{0,0})y) + \frac{1}{\Delta^2}(\lambda_{0,0} - \lambda_{1,0} - \lambda_{0,1} + \lambda_{1,1})xy \tag{60}$$

[00234] Using these assumptions, we follow the same steps as in hereinabove to find Q as a function of G. For instance, this function for one n-pixel as defined in Figure 16(c) is

$$\mathcal{Q}_1 = (\mathbf{CL})^t\mathbf{G}$$

where C, L and G are matrices defined as

$$\mathbf{C} = \begin{bmatrix} 1/24 & 7/24 & 1/24 & 1/24 & 7/24 & 1/24 \\ 0 & 1/24 & 1/48 & 0 & 1/24 & 1/48 \\ 1/48 & 1/24 & 0 & 1/48 & 1/24 & 0 \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \lambda_{0,-1} \\ \lambda_{0,0} \\ \lambda_{0,1} \\ \lambda_{1,-1} \\ \lambda_{1,0} \\ \lambda_{1,1} \end{bmatrix} \quad \text{and} \quad \mathbf{G} = \begin{bmatrix} \mathcal{G}_1^1 \\ \mathcal{G}_3^1 \\ \mathcal{G}_5^1 \end{bmatrix}$$

[00235] Using Equations 40 and 43, we can express D as a function of T . For one 3-pixel, γ_E of $K^{s'}$ as defined in Figure 19, we have

$$< \mathcal{D}, \gamma_E >= (\mathbf{C}_\lambda \mathbf{L}_\lambda)^t\, \mathbf{T}\Delta t \tag{61}$$

where C, L and T are matrices defined as

$$\mathbf{C}_\lambda = \begin{bmatrix} 1/24 & 1/16 & 0 & 1/16 & 1/12 & 0 & 0 & 0 & 0 \\ 1/48 & 1/2 & 1/48 & 0 & 5/24 & 0 & 0 & 0 & 0 \\ 0 & 1/16 & 1/24 & 0 & 1/12 & 1/16 & 0 & 0 & 0 \\ 1/48 & 0 & 0 & 1/4 & 5/24 & 0 & 1/48 & 0 & 0 \\ -1/12 & -3/8 & -1/12 & -3/8 & -7/6 & -3/8 & -1/12 & -3/8 & -1/12 \\ 0 & 0 & 1/48 & 0 & 5/24 & 1/4 & 0 & 0 & 1/48 \\ 0 & 0 & 0 & 1/16 & 1/12 & 0 & 1/24 & 1/16 & 0 \\ 0 & 0 & 0 & 0 & 5/24 & 0 & 1/48 & 1/4 & 1/48 \\ 0 & 0 & 0 & 0 & 1/12 & 1/16 & 0 & 1/16 & 1/24 \end{bmatrix}, \quad \mathbf{L}_\lambda = \begin{bmatrix} \lambda_{-1,-1} \\ \lambda_{0,-1} \\ \lambda_{1,-1} \\ \lambda_{-1,0} \\ \lambda_{0,0} \\ \lambda_{1,0} \\ \lambda_{-1,1} \\ \lambda_{0,1} \\ \lambda_{1,1} \end{bmatrix} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} \mathcal{T}^1_{-1,-1} \\ \mathcal{T}^1_{0,-1} \\ \mathcal{T}^1_{1,-1} \\ \mathcal{T}^1_{-1,0} \\ \mathcal{T}^1_{0,0} \\ \mathcal{T}^1_{1,0} \\ \mathcal{T}^1_{-1,1} \\ \mathcal{T}^1_{0,1} \\ \mathcal{T}^1_{1,1} \end{bmatrix}$$

[00236] We apply Equation 33 with $<S,\gamma_E>$ = 0 and we obtain, for each 3-pixel of $K^{s'}$,

$$\mathcal{T}^1_{0,0} - (\mathbf{C}_\lambda \mathbf{L}_\lambda)^t \, \mathbf{T} \Delta t = \mathcal{T}^0_{0,0}$$

which defines the system of linear equations. The initial conditions are T^0 = I(x).

A Different Hypothesis for Heat Conduction

[00237] In the preceding discussion, we approximated $\lambda(\mathbf{x})$ with a bilinear function, essentially for the sake of simplicity. Nevertheless, it could be preferable to use another assumption. Actually, this simple approach does not accurately handle abrupt changes in conductivity. For example, let us consider two 2-pixels of K^s, as shown in Figure 20. To compute Q, we need to approximate $\lambda(\mathbf{x})$ at the borders of the pixels based on the values at their centers. Using bilinear approximation, the value of $\tilde{\lambda}(\mathbf{x})$ on the line linking two points is declared be the arithmetic mean of the values at these points. For instance, given $\lambda_{0,0} \to 0$ and $\lambda_{1,0} \to 1$, the conduction at the border is about 0.5. This means that the zero conductivity at one pixel is partly cancelled out by the fact that on the pixel beside it, there is a high conductivity coefficient. In non-linear gray level diffusion, we are confronted with precisely this kind of abrupt change. For example, at step edge pixels, the conduction may needs to be very low, whereas immediately adjacent, it may needs to be almost one.

[00238] A better assumption would thus be to consider $\tilde{\lambda}(\mathbf{x})$ as constant over one single 2-pixel of K^s [37]. Therefore, on the 1-face common to two pixels as in Figure 20, we have

$$\tilde{\lambda}(\mathbf{x}) = \left(\frac{0.5}{\lambda_{0,0}} + \frac{0.5}{\lambda_{1,0}}\right)^{-1} = \frac{2\lambda_{0,0}\lambda_{1,0}}{\lambda_{0,0}+\lambda_{1,0}}$$

[00239] It can easily be seen that when $\lambda_{0,0} \to 0$, then $\tilde{\lambda} \to 0$ and when $\lambda_{0,0} << \lambda_{1,0}$, then $\tilde{\lambda} \to 2\lambda_{0,0}$. This means that in both situations, the low conductivity would prevail at the boundary common to the two pixels [37]. With this assumption, the matrices C_λ and L_λ are modified as follows:

$$\mathbf{C}_\lambda = \begin{bmatrix} 1/4 & 1/4 & 0 & 0 \\ 3/2 & -1/4 & -1/4 & 0 \\ 1/4 & 0 & 1/4 & 0 \\ -1/4 & 3/2 & 0 & -1/4 \\ -3/2 & -3/2 & -3/2 & -3/2 \\ -1/4 & 0 & 3/2 & -1/4 \\ 0 & 1/4 & 0 & 1/4 \\ 0 & -1/4 & -1/4 & 3/2 \\ 0 & 0 & 1/4 & 1/4 \end{bmatrix}, \quad \text{and} \quad \mathbf{L}_\lambda = \lambda_{0,0} \begin{bmatrix} \lambda_{0,-1}/(\lambda_{0,-1}+\lambda_{0,0}) \\ \lambda_{-1,0}/(\lambda_{-1,0}+\lambda_{0,0}) \\ \lambda_{1,0}/(\lambda_{1,0}+\lambda_{0,0}) \\ \lambda_{0,1}/(\lambda_{0,1}+\lambda_{0,0}) \end{bmatrix}$$

Experimental Results

[00240] The proposed approach was tested on real and synthetic images in the context of linear isotropic diffusion, optical flow and non-linear diffusion. The results were compared with another method in each case.

[00241] For linear diffusion, Figure 21(a) presents our physics-based method (PB) at three different scales and Figure 21(b) shows the result by convolution for the same scales. In the absence of a quantitative evaluation, we can say that subjectively the results seem to be similar which fulfills our objective of validating our approach.

[00242] For optical flow, Figure 22 shows the first frames of three sequences: the rotating sphere, Hamburg taxi and tree sequences. The results are compared with those being obtained using a finite-difference implementation of the Horn and Schunck algorithm (FD) [18, 19]. In these three examples and for both the PB and FD methods, the image derivatives are computed by convolution with the appropriate Gaussian derivatives. Both temporal and spatial scales are set to 1, as is the weighting factor α . Figure 23 shows the flow pattern computed for the sphere sequence. Figures 24 and 25 present the flow patterns for the taxi and tree sequences, respectively. For the rotating sphere and the taxi sequences, we obtain similar results with both methods. For the tree sequence, we also obtain similar results even if the extreme values seem to be smaller with the PB method than with the FD method. This fact is more apparent in Figure 27(a) and 27(b) where we show respectively the results for the PB and FD methods for the tree sequence in which white noise (standard deviation of 10) has been added (see Figure 26). Another advantage of our method is that it avoids iterations since we apply the algorithm only once.

[00243] For nonlinear diffusion, Figure 28 compares the PB with constant hypothesis on λ and FD [38, 17] methods for a small window of the peppers image with σ = 5. Figure 28(b) presents the original section with white noise added (standard deviation of 10). Figures 28(b) and 28(c) show respectively the results for the PB and FD methods. One can notice that some details are better conserved in Figure 28(c) than in Figure 28(b). This fact is enlightened in Figure 29 which shows a profile of the diagonal line starting at the upper right corner and finishing at the lower left corner.

[00244] Figure 30 presents the results for the peppers image with σ = 1.0, 3.0 and 5.0. The results for PB seem a little sharper than the FD results.

[00245] Figure 32 shows the results for the Lena image (Figure 31(a)) with an added white noise of standard deviation 10 (Figure 31(b)) at two

different scales, $\sigma = 4.0$ and 8.0. Again, the PB method seems to give sharper results at both scales.

[00246] Figure 33 presents details of the Lena results at $\sigma = 8.0$. Figure (b) seems smoother in constant zones but some details are lost. For example, compare the eyes, the trim on the hat and the right of the face.

Conclusion

[00247] An alternative approach to the PDE-based resolution of the diffusion problem was described. The proposed approach differs in two significant ways from with the classical PDE resolution scheme: 1) the image is considered as a cubical complex for which algebraic structures such as chains, cochains, boundaries and coboundaries are defined; and 2) the diffusion problem is decomposed into conservative and constitutive basic laws, each of which is represented by cochains and coboundaries.

[00248] The conservative basic laws are represented without approximation while some approximations are required for the constitutive laws. This means that unlike traditional PDE resolution, for which many approximations must be made, all approximations are known since they are only needed in the representation of the constitutive equations. Coboundaries are computed using fundamental theorems of calculus such as the Green, Stokes and Line Integral theorems. Unlike iterative numerical analysis algorithms that do not allow the explanation of intermediate results, the use of basic laws allows the physical explanation of all steps and intermediate results of the algorithm. Moreover, since there is no iteration in the resolution process, there is no problem about the convergence of the numerical analysis scheme. Furthermore, the use of cubical complexes provides algorithms that can operate in any dimension. It has the significant advantage of avoiding the potentially difficult task of extending the algorithm to higher dimensions. Cochains and coboundaries allow the use of both global and local quantities. Integrals or discrete summations over fields are used to compute global

quantities. This allows the reduction of noise by performing a smoothing operation, as opposed to differentiation, which enhances high frequencies.

[00249] In computer vision and image processing, several problems can be modeled as diffusion problems. The proposed approach has been validated on smoothing by linear and nonlinear diffusion and on the computation of optical flow. The results obtained confirm the effectiveness of this approach.

PRACTICAL EXAMPLE #2:

A PHYSICS-BASED MODEL FOR ACTIVE CONTOURS

[00250] A new active contours model is presented. It is based upon a decomposition or the linear elasticity problem into basic physical laws. As opposite with other physics-based active contours model which solve the partial differential equation arising from the physical laws by some purely numerical techniques, we use exact global values and make approximations only when they are needed. Moreover, these approximations can be made wisely assuming some knowledge about the problem and the domain. The deformations computed with our approach have a physical interpretation. In addition, the deformed curves have some interesting physical properties such as the ability to recover their original shape when the external forces are removed. The physical laws are encoded using the computational algebraic topology based image model described herein. The resultant numerical scheme is then straightforward. The image model allows our algorithm to perform with either 2D or 3D problems.

Introduction

[00251] These last years, active contours and active surfaces have been widely studied since the introduction of active contours by Kass et al [59]. They have been used in image segmentation [62], tracking [68], automatic correction and updating of road databases [46], etc.

[00252] To solve these problems, many different approaches have been proposed (see [57, 63]) in particular physical models derived from equations of continuum mechanics. Mass-springs models are physical models which use a discrete representation of the objects. Objects are modeled as a lattice with point masses linked together by springs [57]. Information is thus only available at a finite number of points [63]. These methods offer only a rough approximation of the phenomena happening in a body [57]. Moreover, the determination of springs constants reflecting the material properties may be a very fastidious work. However, they offer real-time performances and allows for parallel computations.

[00253] Other physical models based upon the minimization of an energy functional which takes into account an internal regularizing force and an external force applied on the data are also often used. Some of them consider the deformable bodies as continuous objects by approximating their continuous behavior with methods such as the finite element method (FEM). FEM are closer to the physics than mass-springs models but their computational requirements make them difficult to be applied in real-time systems without preprocessing steps [57]. Finite difference methods (FDM) are also used to discretize the objects. They usually offer better perfomance than FEM but they require the computation of fourth order derivatives which make them sensitive to noise [63].

[00254] For a given curve S, the application of the FEM and FDM methods leads to a discrete stationary system of equations of the form

$$\mathbb{K}S = \mathfrak{f}(S)$$

where $\mathbb{K}$ is a matrix which encodes the regularizing constraints on S and $\mathfrak{f}$(S) represents the data potential. However, some problems such as animation in graphics applications require to take into account a dynamic evolution of the curve [57]. In these case, inertial body forces and damping forces may also by considered by controlling the deformations through a Newtonian law of motion

$$\mathbf{M}\frac{\partial^2 S}{\partial t^2} + \mathbf{D}\frac{\partial S}{\partial t} + \mathbf{K}S = \mathbf{f}(S) \qquad (62)$$

or by a Lagrangian evolution

$$\frac{\partial S}{\partial t} + \mathbf{K}S = \mathbf{f}(S) \qquad (63)$$

where **M** and **D** are respectively matrices which represent the mass model and the background damping. Equations 62 and 63 are solved using various numerical schemes [70, 64] assuming an initial curve S_0 close to the solution which evolves until the inertial terms go to zero.

[00255] Over the last years, a lot of different methods have been introduced to compute the matrices **M**, **D** and **K** but as pointed out by Montagnat et al [63], these methods have a major drawback: the corresponding system deformations do not have any physical interpretations.

[00256] A new model which includes a physical interpretation of the deformations is described hereinbelow. The model is similar to a mass-springs model but it includes both the efficiency of the mass-springs models and the accuracy of the physical modeling of the FEM by providing a systematic method for specifying springs constants reflecting the properties of the materials.

[00257] To achieve it, we propose to use directly the basic laws of physics which lead to the partial differential equations 62 and 63. These equations are indeed obtained by considering and mixing together some basic laws of physics into a global conservation law and considering its local counterpart [72]. This approach is not always well suited for problems such as computer vision in which the continuous domain must be subdivided into many sub-domains for which there are often only one information available. The use of this information as a global value over each sub-domain allow to directly use

the global conservation law which can lead to an algorithm less sensitive to noise.

[00258] To encode these global values over points, surfaces, volumes, etc arising from some physical laws, we use the computational algebraic topology based image model described hereinabove.

[00259] Our approach has several advantages. 1) Since the linear elasticity problem is well-known in continuum mechanics, our modeling can be made wisely in order to provide some good physical interpretation of the whole deformation process and of its intermediate steps. This allows an easier determination of the parameters used in the process since they have a physical meaning; 2) The determination of the springs constants in order to reflect the material properties is straightforward; 3) The objects in the image (e.g. curves, surfaces) are modeled as entities having their own physical properties such as elasticity and rigidity. They have the property of recovering their original state when the forces applied on them are removed; 4) Both smooth results and results having high curvature points can be obtained; 5) The complexity of the algorithm is minimal and allows for real-time simulation without any extra preprocessing steps [51]; 6) The image model allows our algorithm to perform with either 2D or 3D problems.

Physical Modeling

[00260] One of the objectives of this aspect of the present invention is to model the objects in an image (e.g. curves, surfaces, etc) as entities having their own physical properties such as elasticity and rigidity. As a consequence, these objects need to satisfy the laws and principles of the continuum mechanics. For instance, a body subjected to forces must move or deform according to the universal laws of physics.

[00261] These principles and laws to which all bodies must obey will now be presented. We first introduce the concepts of stress and strain which

are essential in the statement of the governing equations for deformable bodies. We then present the physical laws related to the linear elasticity problem.

[00262] The elasticity theory has been widely studied by engineers and scientists and is the main subject of many books. We only present the concepts of this theory which are relevant to our application. The concepts presented here are well-known and may be found in many continuum mechanics books such as [65, 50].

Forces, Stresses and Strains.

[00263] A material body in a 3-D space is always subjected to forces. These forces may come from an external agent (external forces) or issue from the object itself (internal forces). When the external forces are greater than the internal forces, then the body can undergo deformations (strains) or be accelerated. This deformation can induce internal forces (stresses) if the material is elastic. The concepts of force, stress and strain and the relation between strain and stress is exposed hereinbelow.

Forces acting on a body

[00264] Two basic types of forces act on a body. First, there are the interatomic forces which hold the body's particles together at some configuration. These forces, called internal forces, can either attempt to separate or bring the particles closer according to the fact that the body undergoes a contraction or an extension [65]. They act in response to a force applied by some other agent. Assuming the equilibrium of the body and using Newton's law of reaction, they must be equal in magnitude to the forces applied on the body but in opposite direction [66].

[00265] On the other hand, there are the forces applied by an external agent, called external forces. Two types of these forces are generally applied on a body. First, there are forces such as gravity and inertia called the

body forces, which act on all volume elements. These forces, noted b_i (forces per unit of mass in a direction x_i), are distributed in every part of the body. Secondly, there are forces which act on and are distributed over a surface element such as the contact forces between solid elements [49]. These forces, noted f_i (forces per unit of area in a direction x_i, are called the surface forces. The surface element may be inside the body or a part of a bounding surface [60]. A body of arbitrary size, shape and material subjected to surface and body forces is shown in Figure 34.

[00266] External forces applied on a body must be transmitted to it. A rigid body can then undergo either a spatial shift, a rotation of both of them. In the case of a non-rigid body, it can also go through a deformation or a distortion in which case internal forces will be developed to counterbalance the external forces. If the internal and external forces are balanced, we say that the body is in static equilibrium. Otherwise the body can be accelerated which would give rise to inertia forces [58]. Using d'Alembert's principle, these forces may be included as part of the body forces [48] such that the equilibrium equations can be satisfied. If the body gets deformed then the deformation can either be elastic or not and is subjected to the material properties of the body such as its elasticity and its rigidity. If the internal forces induced by the material properties of a body are too weak to counterbalance the external forces, then the body can be permanently deformed [52].

[00267] We assume herein the material to be isotropic with respect to some mechanical properties. We then suppose that the material properties are the same in all directions for a given point [60]. We also consider an homogeneous material which means that its properties are identical at all locations.

The Concept of Stress at a Point

[00268] Let us consider an isotropic and homogeneous body B. Let us assume that B is subjected to arbitrary surface and body forces such that B

is in static equilibrium. Let P be a interior point of B and S be a plane surface passing through P. S will be referred to as the cutting plane and is defined by the unit normal vector $\mathbf{n}=(n_1, n_2, n_3)^T$. Then S partitions B into two sections I and II as shown in Figure 35.

[00269] Let us assume that ΔS is a small element of area of the cutting plane surrounding P (see Figure 36).

[00270] Since the body is in static equilibrium then the force system acting on each part I and II taken alone must also be in equilibrium. This generally requires that some internal forces are transmitted by part I to part II. These forces are not necessarily distributed uniformly on every part of the cutting plane. Thus they may vary in magnitude and direction over it. We generally want to determine precisely that force distribution at every point of ΔS. The term stress is used to define the intensity and the direction of the internal force Δf acting at point P. Using the Cauchy stress principle [60] we define the stress vector (or traction vector or traction forces) $\mathbf{t}^n$ at P as

$$\mathbf{t}^n = \lim_{\Delta S \to 0} \frac{\Delta f}{\Delta S}$$

assuming that P remains an interior point of ΔS as its area reduces to zero.

[00271] Let us mention that $\mathbf{t}^n$ is not necessarily in the direction of the normal vector **n** at P. However, it may be decomposed into a component perpendicular to the cutting plane, called the normal stress, and a component parallel to it, called the shear stress. The normal stress attempts to separate (bring closer) the material particles after a compression (an extension) of an elastic body when it tries to recover its original state. On the other hand, the shear stress acts parallel to the cutting plane and tends to slide adjacent planes with respect to each other (see Figure 37).

[00272] We must notice that the stress vector $\mathbf{t}^n$ is defined with respect to the cutting plane's unit normal vector **n**. Since there are infinitely

many cutting planes going through P there are also as many stress vectors defined at P. Juvinall [58] defines the state of stress at a point P as a complete description of the stress magnitude and direction for all possible cutting plane passing through P. Fortunately, this description can be fully obtained by considering any three mutually perpendicular planes passing at P [58]. For the sake of simplicity, we usually use the three axes defined by the three canonical vectors x_1, x_2 and x_3.

[00273] Let us define σ_{ij} as the stress component in the direction of x_j when the normal vector is parallel to the axe defined by x_i. If i = j then σ_{ii} represents a normal stress. Otherwise, σ_{ij} is a shear stress. With these conventions, the component t_i^n in the direction of x_i of the traction force t^n depend on the normal stress σ_{ii}, the shear stresses σ_{ji} and σ_{ki} and the normal vector **n** such that (see Figure 38)

$$\begin{aligned} t_i^{\mathbf{n}} &= \sigma_{1i}n_1 + \sigma_{2i}n_2 + \sigma_{3i}n_3 \\ &= \sum_{j=1}^{3} \sigma_{ji}n_j \end{aligned} \tag{64}$$

Equation 64 is known as the Cauchy stress formula.

[00274] Since each of the three coordinates axes involves six , there is a total of nine stress components. However the equilibrium of moments at P [49] gives that only six of these are independent that is $\sigma_{ij} = \sigma_{ji}$ for all i, j = 1, 2, 3. This means that the state of stress at a point is fully determined by σ_{11}, σ_{22}, σ_{33}, σ_{12}, σ_{13} and σ_{23}.

The Concept of Strain at a Point

[00275] Any non rigid body goes through deformations and distortions when subjected to forces. The body can either extend or contract (deformation)

or have a geometric modification of its shape (distortion). Figure 39 presents these concepts.

[00276] The term strain refers to the direction and intensity of the deformation at any given point with respect to a specific plane passing through that point [58]. As for stress, the strain is defined according to a specific cutting plane. The state of strain is defined by Juvinall [58] as the complete definition of the magnitude and direction of the deformation at a given point with respect to a all cutting planes passing through that point.

[00277] As for the state of stress, the description can be obtained by considering any three mutually perpendicular planes passing at P. We can therefore see a great similarity between stress and strain. However, there is a major difference between them: strains are generally some directly measurable quantities while stresses are not. Fortunately, stresses can be computed from strains (and vice versa) using a constitutive equation.

[00278] As for stress, two types of strains can be defined. First, there are the strains which result of a change in the dimensions of the body (deformation). Let B be the same body defined hereinabove, ΔB be a small element of B of length Δx_i in the direction of x_i and $\Delta B'$ be the deformation of ΔB such that Δu_i is the change in lenght of ΔB after the application of a force in the direction of x_i (see Figure 40).The normal strain ε_{ii} at P in the x_i direction with respect to a cutting plane having x_i as normal vector is the unit deformation of a line element [65] in the direction of x_i. It is formally defined as

$$\varepsilon_{ii} = \lim_{\Delta x_i \to 0} \frac{\Delta u_i}{\Delta x_i} = \frac{\partial u_i}{\partial x_i}$$

[00279] The normal strain is clearly the unit change in length per unit original length for the element in the direction of x_i. Since it is the ratio of two units of length, it is dimensionless even if it is sometimes expressed as units of length per unit of length such as inches per inch.

[00280] Let us now suppose that we have two perpendicular lines PB and PA of length Δx_j and Δx_k respectively in the direction of $\mathbf{x}_j$ and $\mathbf{x}_k$ (see Figure 41).

[00281] Let us assume that after a distortion points A and B move respectively to A' and B' while P remains fixed. The lines PA and PB have been rotated of angles θ_{jk} and θ_{kj} such that

$$\frac{\Delta u_j}{\Delta x_k} = \tan(\theta_{jk}) \text{ and } \frac{\Delta u_k}{\Delta x_j} = \tan(\theta_{kj})$$

where Δu_j and Δu_k are respectively the displacements of B and A in the $\mathbf{x}_j$ and $\mathbf{x}_k$ directions.

[00282] If we assume that only small distortions occur, then we can approximate both tangents by their angles. Thus

$$\theta_{jk} \cong \tan(\theta_{jk}) = \frac{\Delta u_j}{\Delta x_k} \text{ and } \theta_{kj} \cong \tan(\theta_{kj}) = \frac{\Delta u_k}{\Delta x_j}$$

or, taking their infinitesimal analogous

$$\theta_{jk} = \lim_{\Delta x_k \to 0} \frac{\Delta u_j}{\Delta x_k} = \frac{\partial u_j}{\partial x_k} \text{ and } \theta_{kj} = \lim_{\Delta x_j \to 0} \frac{\Delta u_k}{\Delta x_j} = \frac{\partial u_k}{\partial x_j} \quad (65)$$

[00283] The shear strain γ_{ik} at P with respect to the cutting plane having $\mathbf{x}_i$ as normal vector is the angle in radians through which two orthogonal lines in the undistorted body are rotated by a distortion [56]. That is $\gamma_{ik} = \theta_{jk} + \theta_{kj}$. The two subscripts in γ_{ik} have a similar meaning as for stress. For instance, γ_{ik} is the strain acting on two adjacent planes perpendicular to the $\mathbf{x}_i$ axis and sliding them relative to each other in the $\mathbf{x}_k$ direction.

[00284] Let us recall that these definitions have been made under the assumption that only very small displacements occur in the body. The normal

and shear strains are supposed small compared to unity [50]. If we relax this constraint in order to include large deformations then the system to solve for the computation of the forces, the stresses, the strains or the displacements becomes non-linear and then harder to solve. This is sometimes necessary in some problems where large deformations can occur such as for thin flexible bodies [50] of for the modelization of human tissue [53]. However, if we restrict ourselves to small deformations, then the approximations made to define the shear strains do not induce too much errors and are widely accepted in the classical theory of elasticity [69, 65, 49, 50].

[00285] Finally, Equation 65 clearly shows that the shear strain is also a dimensionless quantity since it is the ratio of two units of length. If we define for i ≠ j

$$\varepsilon_{ij} = \frac{1}{2}\gamma_{ij} \qquad (i, j = 1, 2, 3)$$

then we get the strain-displacement relations (or the kinematical relationship [69])

$$\varepsilon_{ij} = \frac{1}{2}\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right] \tag{66}$$

[00286] As for stresses, there is a total of nine strain components at each point of the body (three per mutually perpendicular cutting planes) but by symmetry they can be reduced to six, that is $\gamma_{jk} = \gamma_{kj}$ for all j, k = 1, 2, 3 with j ≠ k. The state of strain at any point can then be described by ε_{11}, ε_{22}, ε_{33}, ε_{12}, ε_{13} and ε_{23}.

Relations Between Forces, Stresses and Strains

[00287] As mentioned hereinabove, strains are measurable quantities while stress are not even if both can be computed from the other. This is due to the fact that some knowledge about the material of a body is necessary to

measure the stresses from strains and vice versa. For instance, a steel beam and a rubber beam induce different internal forces when bent equally.

[00288] We will now fill this gap between strains and stresses by stating the physical laws relating them. This hole between strains and stresses need to be filled by a constitutive equation (or material law) which reflect the internal constitution of the materials. The material law for the linear elasticity problem is known as the Hooke's law.

[00289] Before stating the law, let us recall that a material has an elastic behavior when it satisfies the two following conditions:

1. The stresses depend only on the strains.
2. Its properties allow a body to recover its original shape when the external forces applied on the body are removed [60].

[00290] If theses conditions are not satisfied, a body is said to have an inelastic behavior. Any body may be seen as having an elastic behavior as long as it is not deformed beyond a limiting value. This value is called the elastic limit [52, 49] and is usually defined as the maximum value of stress that a body can undergo without permanently being deformed.

[00291] In addition, if the stress is a linear function of the strain we say that an elastic material has a linear elastic behavior. In what follows, we assume that the material has a linear elastic behavior.

[00292] As mentioned in [60] and [49], in many situations the problem of elasticity can be considered as a 2D problem. This particular case is known as plane elasticity. Two basic types of problems compose the plane elasticity. The problems in which the stress components in one direction for a body are all zero are refered to as plane stress problems. On the other hand, if all the strain components in one direction for a body are zero then the state of strain for that body is refered to as plane strain (see [65, 60, 58]).

[00293] We consider our problem as a plane strain problem. This distinction is important since the constitutive equation slightly differs according to the fact that we consider a plane stress or a plane strain problem.

The Hooke's Law

[00294] When a rubber ball is compressed its diameter in the directions perpendicular to the applied force gets larger. A similar phenomenon happen occur when a rubber band is extended and its cross section gets smaller[1]. In fact, these changes in dimensions happen in all materials even if they can't always be noticed by a naked eye [52].

[00295] When a stress is acting on an isotropic and homogeneous body in only one direction (uniaxial stress), one can show that the transverse strain $\varepsilon^{\perp}$ (the strain in a perpendicular direction) is directly proportional to the strain ε induced by the stress

$$\varepsilon^{\perp} = -\nu\varepsilon$$

The ratio

$$\nu = \frac{-\varepsilon^{\perp}}{\varepsilon}$$

is called the Poisson ratio. It is supposed constant when the stress is below the elastic limit [66].

[00296] The linear relationship between a uniaxial stress σ_{ii} in a direction x_i and the corresponding strain ε_{ii} is known as the Hooke's law [52, 58] and is written as

$$\varepsilon_{ii} = \frac{\sigma_{ii}}{E}$$

[1] Example taken in [52]

where E is the Young's modulus of elasticity. Of course, the Hooke's law is valid if the stress is not beyond the elastic limit of the material.

[00297] As pointed out by Boresi [49], the stresses at any point depend on all the strains in the neighborhood of that point. Thus the total deformation in the x_i direction depends not only on the stress in that direction but also of the deformations in the other two perpendicular directions. For instance the normal strain ε_{ii} does not only depend on $\frac{\sigma_{ii}}{E}$ (Hooke's law) but also of the transverse strains ε_{jj} and ε_{kk} such that the total deformation in the direction of x_i is

$$\begin{aligned}\varepsilon_{ii} &= \frac{\sigma_{ii}}{E} - \nu\varepsilon_{jj} - \nu\varepsilon_{kk} \\ &= \frac{\sigma_{ii}}{E} - \nu\frac{\sigma_{jj}}{E} - \nu\frac{\sigma_{kk}}{E} \\ &= \frac{1}{E}\left[\sigma_{ii} - \nu\left(\sigma_{jj} + \sigma_{kk}\right)\right] \quad (i \neq j \neq k)\end{aligned} \tag{67}$$

[00298] Equation 67 is the normal strain-stress relation. For isotropic materials, it can be shown that the normal strains are not influenced by the shear stresses [52]. Consequently, the shear stresses only induce shear strains and they are related by the relation

$$2\varepsilon_{ij} = \frac{\sigma_{ij}}{G} \quad (i \neq j) \tag{68}$$

where $G = \frac{E}{2(1+\nu)}$ is called the modulus of rigidity. Equations 67 and 68 are known as the Generalized Hooke's law for linear elastic isotropic materials [52]. These equations can be inverted in order to express the stresses as functions of the strains

$$\sigma_{ii} = \frac{E}{(1+\nu)(1-2\nu)}[(1-\nu)\varepsilon_{ii} + \nu(\varepsilon_{jj}+\varepsilon_{kk})] \tag{69}$$

$$\sigma_{ij} = 2G\varepsilon_{ij} = \frac{E}{1+v}\varepsilon_{ij} \quad (i \neq j) \tag{70}$$

Relation Between Forces and Stresses.

[00299] It is well known that the conservation (balance, equilibrium) laws constitute an important class of equations in continuum mechanics. They relate the change in total amount of a physical quantity inside a body with the amount of this quantity which flows through its boundary. These laws must be satisfied for every continuous materials. Local differential equations are usually used to express these laws. In what follows, we present the linear momentum which is relevant for the linear elasticity problem.

[00300] To every material body B is associated a measure of its inertia called the mass. This measure may vary in space and time inside a body. Let V be the volume of B, S its bounding surface and Δm be the mass of a small amount of volume ΔV. We call mass density

$$\rho = \rho(\mathrm{x}, t) = \lim_{\Delta V \to 0} \frac{\Delta m}{\Delta V}$$

[00301] Let us assume that distributed body forces ρb_i and tractions forces t_i^n are applied to S (see Figure 42). Let also assume that B is moving under the velocity field $v_i = v_i(x,t)$. The quantity

$$P_i(t) = \iiint\limits_V \rho v_i \, dV$$

is called the linear momentum of B. The principle of linear momentum [49] states that the resultant force acting on a body is equal to the time rate of change of the linear momentum. Thus

$$\frac{d}{dt}\iiint\limits_V \rho v_i \, dV = \underbrace{\iint\limits_S t_i^n \, dS + \iiint\limits_V \rho b_i \, dV}_{\text{Forces acting on the body}} \qquad (71)$$

[00302] Recalling Equation 64

$$t_i^n = \sum_{j=1}^{3} \sigma_{ji} n_j$$

where $\mathbf{n} = (n_1, n_2, n_3)^T$ is the unit normal vector to the surface and $\sigma = (\sigma_{1i}, \sigma_{2i}, \sigma_{3i})^T$ and using Gauss's divergence theorem, we have

$$\frac{d}{dt}\iiint\limits_V \rho v_i \, dV = \iiint\limits_V \rho \frac{\partial v_i}{\partial t} \, dV = \iiint\limits_V (\nabla \cdot \sigma + \rho b_i) \, dV \qquad (72)$$

[00303] Since V is arbitrary then the integral sign can be retrieved leading to the local equations of motion

$$\rho \frac{dv_i}{dt} = \nabla \cdot \sigma + \rho b_i \qquad (73)$$

[00304] The global equilibrium equations can be obtained assuming a zero velocity field in Equation 72

$$\underbrace{\iiint\limits_V \nabla \cdot \sigma \quad dV}_{\text{Internal forces}} + \underbrace{\iiint\limits_V \rho b_i \quad dV}_{\text{External forces}} = 0 \qquad (i = 1, 2, 3) \qquad (74)$$

and their local counterparts are

$$\nabla \cdot \sigma + \rho b_i = 0 \qquad (i = 1, 2, 3) \qquad (75)$$

[00305] Let us now summarize the decomposition of the problem. We introduced the relation between the global displacement **U** of a body and the corresponding strains (see Figure 43(a)) hereinabove. We also presented the constitutive equation relating the strains and the stresses (see Figure 43(b)). Finally, we described how the stresses are related to forces using the linear momentum principle (see Figure 43(c)). A general scheme similar to the one presented by Tonti [72] may then be introduced to summarize how the internal reaction forces of a body are related to the global displacements of that body (Figure 44).

Discrete Representation of Images

[00306] Some algebraic tools used to model the image will now be recalled from the above description. An image is composed of two distinctive parts: the image support (pixels) and some field quantity associated to each pixel. This quantity can be scalar (e.g. gray level), vectorial (e.g. color, multispectral, optical flow) or tensorial (e.g. Hessian). We model the image support in terms of cubical complexes, chains and boundaries. With these concepts, we are able to give a formal description of an image support of any dimension. For quantities, the concept of cochains which are representations of fields over a cubical complex is introduced. For the use of these concepts in image processing, see [45].

[00307] An image is a complex of unit cubes usually called pixels. A pixel $\gamma \subset \Re^n$ is a product

$$\gamma = I_1 \times I_2 \times \ldots \times I_n$$

where I_i is either a singleton or an interval of unit length with integer endpoints. Then I_i is either the singleton {k} and is said to be a degenerate interval, or the closed interval [k, k+1] for some $k \in Z$. The number $q \in \{0,1,\ldots,n\}$ of non-degenerate intervals is by definition the dimension of γ which is called a q-pixel. Figure 45 illustrates three elementary pixels in $\Re^2$.

[00308] For $q \geq 1$, let $J = \{k_0, k_1, \ldots, k_{q-1}\}$ be the ordered subset {1, 2, ..., n} of indices for which $I_{k_j} = [a_j, b_j]$ is non-degenerate. Define

$$A_{k_j}\sigma = I_1 \times \ldots I_{k_j-1} \times \{a_j\} \times I_{k_j+1} \times \ldots \times I_n$$

and

$$B_{k_j}\sigma = I_1 \times \ldots I_{k_j-1} \times \{b_j\} \times I_{k_j+1} \times \ldots \times I_n$$

[00309] The A_{k_j} and the B_{k_j} are called the (q-1)-faces of σ. One can define the (q-2)-faces, ..., down to the 0-faces of σ the same way. The faces of γ different from γ itself are called its proper faces.

[00310] By definition, a natural orientation of the cube is assumed for each pixel. Suppose that γ denotes a particular positive oriented q-pixel. It is natural to denote the same pixel with opposite orientation by $-\gamma$. Examples of orientations are given in Figure 45. A cubical complex in $\Re^n$ is a finite collection K of q-pixels such that every face of any pixel of the image support called K is also a pixel in K and the intersection of any two pixels of K is either empty or a face of each of them. For example, traditional 2D image models was only considering pixel as a 2D squared element. Definitions presented before allows us to consider 2-pixels (squared elements), 1-pixels (line elements) and 0-pixels (punctual elements) simultaneously.

[00311] In order to write the image support in algebraic form, we introduce the concept of chains. Any set of oriented q-pixels of a cubical complex can be written in algebraic form by attributing them the coefficient 0,1 or −1, if they are not in the set or if they should or not be taken with positive orientation, respectively. In order to represent weighted domains, we must allow arbitrary integer multiplicity for each q-pixel.

[00312] Given a topological space $X \subset \Re^n$ in terms of a cubical complex, we get a free abelian group $C_q(X)$ generated by all the q-pixels. The

elements of this group are called q-chains and they are formal linear combinations of q-pixels [45]. A formal expression for a q-chain c_q is $c_q = \sum_{\gamma_i \in K} \lambda_i \gamma_i$ where $\lambda_i \in Z$.

[00313] The last step needed for the description of the image plan is the introduction of the concept of boundary of a chain. Given a q-pixel γ, we define its boundary $\partial\gamma$ as the (q-1)-chain corresponding to the alternating sum of its (q-1)-faces. The sum is taken according to the orientation of the (q-1)-faces with respect to the orientation of the q-pixel. We say that a (q-1)-face of γ is relatively positively oriented with respect to the orientation of γ if its orientation is compatible with the orientation of γ. By linearity, the extension of the definition of boundary to arbitrary q-chains is expedient. For instance, in Figures 45(b) and 45(c), the boundary of the 1-pixel a is $x_2 - x_1$ and the boundary of the 2-pixel A is $a+b-c-d$ and we say that a and b are positively oriented with respect to orientation of A but c and d are negatively oriented with respect to orientation of A. Let us notice that the boundary of a 1-pixel is always the difference of its boundary points. The boundary can be defined recursively. Suppose a (q-1)-chain and a q-chain γ_q defined as $\gamma_q = \gamma_{q-1} \times [a,b]$, the boundary of γ_q can be recursively written as

$$\partial\gamma_q = \partial\gamma_{q-1} \times [a,b] + (-1)^{(q-1)}(\gamma_{q-1} \times \{b\} - \gamma_{q-1} \times \{a\}) \qquad (76)$$

[00314] In order to model the pixels quantity over the image plane, we are looking for an application F which associates a global quantity with all q-pixels γ of a cubical complex. We note this $< F, \gamma >$. This quantity may be any mathematical entities such as scalars, vectors, etc. For two adjacent q-pixels γ_1 and γ_2, F must satisfy $< F, \lambda_1\gamma_1 + \lambda_2\gamma_2 >= \lambda_1 < F, \gamma_1 > + \lambda_2 < F, \gamma_2 >$ which means that the sum of the quantity over each pixel is equal to the quantity over the two pixels. The resulting transformation $F : C_q(X) \rightarrow \Re$ is called a q-cochain and is used as a representation of a quantity over the cubical complex.

[00315] We finally need an operator which associates a global quantity to the (q+1)-pixels according to the global quantities given on their q-faces. Given a q-cochain F, we define an operator ∂, called the coboundary operator, which transforms F into a (q+1)-cochain ∂F such that

$$< \delta\mathcal{F}, \gamma > = < \mathcal{F}, \partial\gamma > \tag{77}$$

for all (q+1)-chains γ. The coboundary is defined as the signed sum of the physical quantities associated with the q-faces of γ. The sum is taken according to the relative orientation of the q-faces of the (q+1)-pixels of γ with respect to their orientation. Figure 46 presents an example of the coboundary operation for a 2-pixel.

Representation of the Equilibrium Equation

[00316] The basic laws of Figure 44 with concepts of algebraic topology in order to get a generic algorithm for solving the equilibrium equation 74 will now be modeled.

[00317] The algorithm is resumed as follows: 1) The image support is firstly subdivided into cubical complexes; 2) Global quantities are computed over pixels of various dimensions via cochains according to basic laws; 3) The constitutive equations 69 and 70 are expressed as a linear transformation between two cochains.

The Relative Displacement

[00318] Let B be a body in a 3D space and K^p be a 3-complex representing the subdivided spatial support of B. Let us consider a 0-cochain $\mathcal{U}$ and a 1-cochain D such that D is the coboundary of $\mathcal{U}$

$$\begin{array}{rcl} \mathcal{D} : \mathcal{C}_1(\mathcal{K}^p) & \rightarrow & \mathbf{R}^3 \\ \gamma & \mapsto & < \mathcal{D}, \gamma > = < \delta\mathcal{U}, \gamma > = < \mathcal{U}, \partial\gamma > \end{array} \tag{78}$$

[00319] Figure 47 presents some examples of $\mathcal{U}$ and D for a 3-pixel of K^p,

[00320] We must now specify the computational rules for both cochains $\mathcal{U}$ and D. Recalling the strain-displacement relation (Equation 66)

$$\varepsilon_{ij} = \frac{1}{2}\left[\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right] \tag{79}$$

we have an application ε'

$$\begin{aligned} \varepsilon' : \mathbb{R}^3 &\rightarrow \mathbb{R}^6 \\ \mathbf{U} &\mapsto \varepsilon'(\mathbf{U}) = (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \varepsilon_{12}, \varepsilon_{13}, \varepsilon_{23})^T \end{aligned}$$

[00321] Omitting the shear strain components as in [67], we may define

$$\begin{aligned} \varepsilon : \mathbb{R}^3 &\rightarrow \mathbb{R}^3 \\ \mathbf{U} &\mapsto \varepsilon(\mathbf{U}) = (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33})^T = \nabla\mathbf{U} \end{aligned} \tag{80}$$

[00322] Using the global form of Equation 80 over a 1-pixel γ_D such that $\partial\gamma_D = x_* - x_\#$, we have

$$\int_{\gamma_D} \varepsilon(\mathbf{U})\, d\gamma_D = \int_{x_\#}^{x_*} \varepsilon(\mathbf{U})\, d\gamma_D = \int_{x_\#}^{x_*} \nabla\mathbf{U}\, d\gamma_D \tag{81}$$

where $d\gamma_D$ is an infinitesimal part of the domain γ_D. Since $\nabla\mathbf{u}$ is a conservative field, we can apply the line integral theorem [55, 71] which states that for a conservative field **F(x)** = ∇f(**x**) and for two points A and B in an opened connected region containing **F(x)** the integral of the tangential part of **F(x)** along a curve R joining A and B is independent of the path

$$\int_A^B \mathbf{F}(\mathbf{x})\, dR = f(B) - f(A)$$

[00323] From Equation 81, we then have

$$\int_{x_\#}^{x_*} \nabla \mathbf{U}\, d\gamma_D = \mathbf{U}(x_*) - \mathbf{U}(x_\#) \tag{82}$$

[00324] On the other hand, applying the cochain D to the 1-pixel γ_D we also have

$$<\mathcal{D},\gamma_D>=<\mathcal{U},\partial\gamma_D>=\mathcal{U}(x_* - x_\#)=\mathcal{U}(x_*)-\mathcal{U}(x_\#) \tag{83}$$

which is the same form as Equation 82. We then define U(x) = **U(x)** Consequently, the location of the displacement vector **U** must correspond to the 0-pixels of K^p. The previous definitions are extended by linearity to the 1-chains of K^p.

The Force-Stress Relation

[00325] Let us consider another 3-complex K^s also representing the subdivided spatial support of the body B. Let us also consider a 3-cochain $\mathcal{F}$ and a 2-cochain S such that $\mathcal{F}$ is the coboundary of S

$$\begin{array}{rcl}\mathcal{F}:C_3(\mathcal{K}^p) & \to & \mathbb{R}^3\\ \gamma & \mapsto & <\mathcal{F},\gamma>=<\delta\mathcal{S},\gamma>=<\mathcal{S},\partial\gamma>\end{array} \tag{84}$$

[00326] Figure 48 presents some examples of $\mathcal{F}$ and S for a 3-pixel of K^p,

[00327] Let γ_F be a 3-pixel of K^s and γ_S be a 2-chain over K^s such that $\gamma_S = \partial\gamma_F$. Let us assume that the 2-faces γ_{S_j} of γ_F are relatively positively oriented with respect to the orientation of γ_F. By the definition of the coboundary we have

$$<\mathcal{F},\gamma_F>=\sum_{\gamma_{s_j}}<\mathcal{S},\gamma_{S_j}> \tag{85}$$

[00328] Again, we want to determine the computational rules associated with F and S. If we set a zero velocity field in Equation 71, we have

$$\iint\limits_{V} \rho b_i \, dV = \iint\limits_{S} -\sigma_i \cdot \mathbf{n} \, dS$$

where $\sigma_i = (\sigma_{1i}, \sigma_{2i}, \sigma_{3i})$. To fulfill Equation 85 we define

$$< \mathcal{F}_i, \gamma_F >= \iiint\limits_{V} \rho b_i \, dV \qquad (86)$$

and

$$< \mathcal{S}_i, \gamma_{S_j} >= \iint\limits_{S} -\sigma_i \cdot \mathbf{n} \, dS \qquad (i = 1, 2, 3) \qquad (87)$$

where

$\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3)^T$

and

$\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)^T$

The Stress-Strain Relation

[00329] We have presented exact global versions of Equations 66 on K^p and 74 on K^s by the means of Equations 83, 86 and 87. In order to complete the scheme of Figure 44, we need to represent the Hooke's law (Equations 69 and 70) which links the local values of $\varepsilon(\mathbf{x})$ and $\sigma(\mathbf{x})$. Since Equations 69 and 70 are constitutive equations, we can't provide a topological expression of them. Instead we express them as linear transformations between the cochains D and S

$$\mathcal{D} \overset{\Delta}{\mapsto} \mathcal{S}$$

[00330] To find this transformation, let us recall Equation 87

$$< \mathcal{S}_i, \gamma_{\mathcal{S}_j} >= \iint_S -\sigma_i \cdot \mathbf{n}\, dS \qquad (i = 1, 2, 3)$$

which links the cochain S with the strains ε using the generalized Hooke's law. Unfortunately, the strains are only known at a finite number of points and must then be approximated over the whole domain S with an approximation function $\tilde{\varepsilon}(\mathbf{x})$. We choose $\tilde{\varepsilon}(\mathbf{x})$ such that for each 1-face γ_D of a 2-pixel γ of K^p we have

$$\int_{\gamma_D} \tilde{\varepsilon}(\mathbf{x}) \cdot dR =< \mathcal{D}, \gamma_D > \tag{88}$$

where dR is an infinitesimal part of the domain represented by γ_D. Let us notice that we only need to approximate the normal components of ε. In fact, if we have

$$\nabla\tilde{\mathbf{U}}(\mathbf{x}) = \left(\frac{\partial \tilde{U}_1}{\partial x_1}(\mathbf{x}), \frac{\partial \tilde{U}_2}{\partial x_2}(\mathbf{x}), \frac{\partial \tilde{U}_3}{\partial x_3}(\mathbf{x})\right)^T = (\tilde{\varepsilon}_{11}(\mathbf{x}), \tilde{\varepsilon}_{22}(\mathbf{x}), \tilde{\varepsilon}_{33}(\mathbf{x}))^T = \tilde{\varepsilon}(\mathbf{x})$$

where $\tilde{u}(\mathbf{x})$ is the approximated displacement vector over γ and if $\tilde{\varepsilon}(\mathbf{x})$ is chosen to satisfy Equation 88, then the vector $\tilde{U}(\mathbf{x})$ is fully determined. Then the shear components of $\tilde{\varepsilon}(\mathbf{x})$ can be computed by appropriately differentiating the components of $\tilde{U}(\mathbf{x})$. Using this remark and applying the generalized Hooke's law to $\tilde{\varepsilon}(\mathbf{x})$ satisfying Equation 88, we have

$$\begin{aligned} \tilde{\sigma}_{ii}(\mathbf{x}) &= \frac{E}{(1+\nu)(1-2\nu)} [(1-\nu)\tilde{\varepsilon}_{ii}(\mathbf{x}) + \nu(\tilde{\varepsilon}_{jj}(\mathbf{x}) + \tilde{\varepsilon}_{kk}(\mathbf{x}))] \\ \tilde{\sigma}_{ij}(\mathbf{x}) &= \frac{E}{(1+\nu)} \tilde{\varepsilon}_{ij}(\mathbf{x}) \qquad (i, j, k = 1, 2, 3) \end{aligned}$$

at all point of γ. Equation 87 is then replaced by

$$< S_i, \gamma_{S_j} >= \iint_S -\widetilde{\sigma}_i(\mathbf{x}) \cdot \mathbf{n}\, dS = \Lambda_i(\varepsilon) \qquad (i = 1, 2, 3) \tag{89}$$

which depends on the choice of the approximation function $\widetilde{\varepsilon}(\mathbf{x})$ and of the relative position of K^s with respect to K^p.

Summary of the Algorithm

[00331] We now summarize the algorithm used to find an expression of the internal forces according to the displacements of a body. The input data for this algorithm are the cochain U and the material properties of the body (the values of E and ν).

1. Choice of the location of K_p with respect to K^s.
2. Computation of the cochain D.
3. Computation of the cochain S
   (a) Choice of the approximation function $\widetilde{\varepsilon}(\mathbf{x})$.
   (b) Application of Equation 89 to express S as a function of the displacement components.
4. Computation of the force by applying Equation 85.

Applications.

Active Contours

[00332] We apply the above described approach to a 2D active contour model based upon a Lagrangian evolution of the curve S

$$\frac{\partial S}{\partial t} + \mathbf{K}S = \mathbf{F}_{ext}(S) \tag{90}$$

where **K** is the matrix which contains the regularization forces of the curve.

[00333] This dynamic system is discretized in time using a finite difference scheme. For a given time step Δt, we can approximate the time derivative by

$$\frac{\partial S}{\partial t} \cong \frac{S_{t+\Delta t} - S_t}{\Delta t}$$

[00334] The curve deformation is governed by each vertex displacements compared to their neighbors until the equilibrium between the inertia forces, the image forces and the internal forces. We solved Equation 90 using an explicit scheme

$$S_{t+\Delta t} = S_t + \Delta t(\mathbf{F}_{ext} - \mathbf{K}S_t) \tag{91}$$

[00335] Assuming that the initial curve S_0 was in an equilibrium state and that the initial body forces $F_0 = KS_0$ are constant during the deformation process, we can add these forces to the external forces F_{ext} leading to a modified version of Equation 91

$$\begin{aligned} S_{t+\Delta t} &= S_t + \Delta t(\mathbf{F}_{ext} + \mathbf{F}_0 - \mathbf{K}S_t) \\ &= S_t + \Delta t(\mathbf{F}_{ext} - (\mathbf{K}S_t - \mathbf{K}S_0)) \\ &= S_t + \Delta t(\mathbf{F}_{ext} - \mathbf{KU}) \end{aligned} \tag{92}$$

where **U** is the displacement vector of the curve S.

[00336] The image subdivision process is similar to the one presented in [57]. We want to solve Equation 92 for local **U(x)** located at the center of each pixel and known initial curve S_0 closed to the solution. Following the steps presented hereinabove, we first position the two dimensional cubical complexes K^p and K^s. As mentioned, K^p must be placed in order to have its 0-pixels corresponding to the center of the image pixels. We positioned K^s in such a way that its 2-pixels coincide with the image pixels. Thus, the 2-pixels of K^s are rectangular and symmetrically staggered with the 1-pixels of K^p and each 1-pixel of intersects orthogonally in the middle of a 1-pixel of K^p.

Mattiussi [61] showed that this way of positioning K^s allows the use of lower order approximation polynomials without losing accuracy. Figure 49 shows the two complexes positions for a 5×5 image.

[00337] In order to solve Equation 92, we need global values **F** over each pixel of K^s. Since these values are generally known, we did not try to express them in a topological way. In our examples, we use the gradient field of the bright line plausibility image obtained using a line detector proposed in [54]. We assume that the gradient provides global values valid over the whole pixel. Thus we set **F** = ∇**L** where **L** is the line plausibility image, ∇**L** = g'_σ\***L** and g'_σ is the Gaussian derivative at scale σ. We also choose an approximation function $\tilde{\varepsilon}$(**x**)=($\tilde{\varepsilon}_{11}$(**x**), $\tilde{\varepsilon}_{22}$(**x**))^T. For simplicity, we assume that ($\tilde{\varepsilon}$(**x**) = $\nabla\tilde{U}$(**x**) arises from a bilinear approximation

$$\tilde{\mathbf{U}}(\mathbf{x}) = (\tilde{U}_1(\mathbf{x}), \tilde{U}_2(\mathbf{x}))^T = \mathbf{a} + \mathbf{b}x_1 + \mathbf{c}x_2 + \mathbf{d}x_1x_2$$

[00338] Thus we have

$$\begin{aligned} \tilde{\varepsilon}_{11}(\mathbf{x}) &= \mathbf{b} + \mathbf{d}x_2 \\ \tilde{\varepsilon}_{22}(\mathbf{x}) &= \mathbf{c} + \mathbf{d}x_1 \end{aligned}$$

[00339] Since $\tilde{\varepsilon}_{11}$(**x**) and $\tilde{\varepsilon}_{22}$(**x**) must satisfy Equation 88 for all 1-faces γ_D of any 2-pixel γ of K^p as in Figure 50, the following relations must hold

$$\begin{aligned} \mathcal{D}_1 &= \int_0^\Delta \tilde{\varepsilon}(x_1, 0) \cdot \vec{\mathbf{i}}\, dx_1 \\ \mathcal{D}_2 &= \int_0^\Delta \tilde{\varepsilon}(\Delta, x_2) \cdot \vec{\mathbf{j}}\, dx_2 \\ \mathcal{D}_3 &= \int_0^\Delta \tilde{\varepsilon}(x_1, \Delta) \cdot \vec{\mathbf{i}}\, dx_1 \end{aligned}$$

$$\mathcal{D}_4 \;=\; \int_0^{\Delta} \tilde{\varepsilon}(0,\Delta) \cdot \overrightarrow{\mathbf{j}}\, dx_2$$

from which we obtain

$$\tilde{\varepsilon}(\mathbf{x}) = \frac{1}{\Delta}\left[\left(\mathcal{D}_1 + \frac{\mathcal{D}_3 - \mathcal{D}_1}{\Delta}x_2\right), \left(\mathcal{D}_4 + \frac{\mathcal{D}_2 - \mathcal{D}_4}{\Delta}x_1\right)\right] = \nabla\tilde{\mathbf{U}}(\mathbf{x}) \tag{93}$$

[00340] From Equation 93 and the definition of the normal strains, it is straightforward that

$$\tilde{\mathbf{U}}(\mathbf{x}) = k + \frac{1}{\Delta}(\mathcal{D}_1 x_1 + \mathcal{D}_4 x_2) + \frac{1}{\Delta^2}(\mathcal{D}_3 - \mathcal{D}_1 + \mathcal{D}_2 - \mathcal{D}_4)\, x_1 x_2 \tag{94}$$

where $k = \tilde{U}(0)$. Using Equations 83 and 94 we have

$$\begin{aligned}\tilde{\mathbf{U}}(\mathbf{x}) = \tilde{\mathbf{U}}(x_1,x_2) \;&=\; \mathbf{U}(0,0) + \frac{1}{\Delta}(\mathbf{U}(\Delta,0) - \mathbf{U}(0,0))\,x_1 + \frac{1}{\Delta}(\mathbf{U}(0,\Delta) - \mathbf{U}(0,0))\,x_2 \\ &+\frac{1}{\Delta^2}(\mathbf{U}(0,0) + \mathbf{U}(\Delta,\Delta) - \mathbf{U}(0,\Delta) - \mathbf{U}(\Delta,0))\,x_1x_2\end{aligned} \tag{95}$$

from which we can deduce the values of $\tilde{\sigma}_i(\mathbf{x}) = (\tilde{\sigma}_{1i}(\mathbf{x}),\ \tilde{\sigma}_{2i}(\mathbf{x}))^T$.

[00341] The last step is the computation of the internal forces **F** for each 2-pixel of K^s. With K^p and K^s positioned as mentioned before, we have that each 2-pixel γ_F of K^s intersects four 2-pixels γ_A, γ_B, γ_C and γ_D of K^p. That is, we must consider four approximation functions $\tilde{\sigma}_i^A$, $\tilde{\sigma}_i^B$, $\tilde{\sigma}_i^C$ and $\tilde{\sigma}_i^D$ corresponding to the four intersecting 2-pixels of K^p (see Figure 51)

[00342] We find the value of the cochain S over the four 1-face of γ by the appropriate integration

$$\begin{aligned}\mathcal{S}_i^1 \;&=\; \int_0^{\frac{\Delta}{2}} \tilde{\sigma}_i^A\left(\frac{\Delta}{2},x_2\right) \cdot \overrightarrow{\mathbf{i}}\, dx_2 + \int_{-\frac{\Delta}{2}}^{0} \tilde{\sigma}_i^B\left(\frac{\Delta}{2},x_2\right) \cdot \overrightarrow{\mathbf{i}}\, dx_2 \\ \mathcal{S}_i^2 \;&=\; \int_0^{\frac{\Delta}{2}} \tilde{\sigma}_i^B\left(x_1,-\frac{\Delta}{2}\right) \cdot -\overrightarrow{\mathbf{j}}\, dx_1 + \int_{-\frac{\Delta}{2}}^{0} \tilde{\sigma}_i^C\left(x_1,-\frac{\Delta}{2}\right) \cdot -\overrightarrow{\mathbf{j}}\, dx_1\end{aligned}$$

$$\begin{aligned} \mathcal{S}_i^3 &= \int_0^{\frac{\Delta}{2}} \widetilde{\sigma}_i^A\left(x_1, \frac{\Delta}{2}\right) \cdot \overrightarrow{\mathbf{j}}\, dx_1 + \int_{-\frac{\Delta}{2}}^0 \widetilde{\sigma}_i^D\left(x_1, \frac{\Delta}{2}\right) \cdot \overrightarrow{\mathbf{j}}\, dx_1 \\ \mathcal{S}_i^4 &= \int_0^{\frac{\Delta}{2}} \widetilde{\sigma}_i^D\left(-\frac{\Delta}{2}, x_2\right) \cdot -\overrightarrow{\mathbf{i}}\, dx_2 + \int_{-\frac{\Delta}{2}}^0 \widetilde{\sigma}_i^C\left(-\frac{\Delta}{2}, x_2\right) \cdot -\overrightarrow{\mathbf{i}}\, dx_2 \end{aligned}$$

[00343] Using Equation 85 we have

$$< F_i, \gamma_F >= \mathcal{S}_i^1 + \mathcal{S}_i^2 + \mathcal{S}_i^3 + \mathcal{S}_i^4 \quad (96)$$

[00344] Substituting Equations 96 and 95 in Equation 96, we can express the internal forces **F** as a function of the displacement **U**. As an example, we present the values of **F** for the 2-pixel γ_F of Figure 50 with $\Delta = 1$

$$\begin{aligned} F_1 &= C\,[(3-4\nu)u_{-1,1} + (2-8\nu)u_{0,1} + (3-4\nu)u_{1,1} + (10-8\nu)u_{-1,0} + (-36+48\nu)u_{0,0} \\ &\quad +(10-8\nu)u_{1,0} + (3-4\nu)u_{-1,-1} + (2-8\nu)u0,-1 + (3-4\nu)u_{1,-1} - 2v_{-1,1} \\ &\quad +2v_{1,1} + 2v_{-1,-1} - 2v_{1,-1}] \\ F_2 &= C\,[(3-4\nu)u_{-1,1} + (10-8\nu)u_{0,1} + (3-4\nu)u_{1,1} + (2-8\nu)u_{-1,0} + (-36+48\nu)u_{0,0} \\ &\quad +(2-8\nu)u_{1,0} + (3-4\nu)u_{-1,-1} + (10-8\nu)u_{0,-1} + (3-4\nu)u_{1,-1} - 2v_{-1,1} \\ &\quad +2v_{1,1} + 2v_{-1,-1} - 2v_{1,-1}] \end{aligned}$$

where

$$C = \frac{E}{16(1+\nu)(1-2\nu)}$$

and

$$\mathcal{U} = \begin{bmatrix} u \\ v \end{bmatrix}$$

[00345] Equation 97 induces a linear relationship between a pixel and its neighbors. This relation is used to build the stiffness matrix of Equation 91. If we let U_{x_i} (i=1, 2) be the displacement vector for the component xi then we have

$$F_i(\mathbf{x}) = U_{x_1} * N_{x_1}^i + U_{x_2} * N_{x_2}^i$$

where

$$N^i_{x_1} = \frac{E}{16(1+\nu)(1-2\nu)} \begin{bmatrix} 3-4\nu & 2-8\nu & 3-4\nu \\ 10-8\nu & -36+48\nu & 10-8\nu \\ 3-4\nu & 2-8\nu & 3-4\nu \end{bmatrix}$$

$$N^i_{x_2} = \frac{E}{16(1+\nu)(1-2\nu)} \begin{bmatrix} -2 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & -2 \end{bmatrix}$$

$$N^i_{x_3} = \frac{E}{16(1+\nu)(1-2\nu)} \begin{bmatrix} -2 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & -2 \end{bmatrix}$$

$$N^i_{x_4} = \frac{E}{16(1+\nu)(1-2\nu)} \begin{bmatrix} 3-4\nu & 10-8\nu & 3-4\nu \\ 2-8\nu & -36+48\nu & 2-8\nu \\ 3-4\nu & 10-8\nu & 3-4\nu \end{bmatrix}$$

[00346] The pairs $(N^i_{x_1}, N^i_{x_2})$, (i = 1, 2) will be referred to as the stiffness kernels.

Computation of the Displacement Vector

[00347] The assumption made when calculating the displacement vector will now be explained. Let v be a vertex of s subdivided curve S and v' be its corresponding vertex in the deformed curve S'. Let us denote by U[v] the entry in the displacement vector **U** corresponding to v. Let us suppose that the displacement is constant in each direction everywhere that is U[v] = $(k_1, k_2)^T$ with $k_1, k_2 \in \Re$ for all v in S. Since the sum of all entries of either $N^1_{x_1}$, $N^1_{x_2}$, $N^2_{x_1}$ or $N^2_{x_2}$ is zero, it follows that $F_1[v] = F_2[v] = 0$ for all v in S which means that there is no internal force induced. The computation of the internal forces with the stiffness kernels is then invariant with respect to translation.

[00348] Let v_1, v_2, v_3, v_4 and v_5 be five adjacent vertices of S and v'_1, v'_2, v'_3, v'_4, v'_5 be their corresponding vertices in S' (see Figure 52).

[00349] Let v_{i,x_j} be the x_j coordinate of the spatial representation of the vertex v_i. Then we have

$$U_{x_i} = \left(\dots, v'_{1,x_i} - v_{1,x_i}, v'_{2,x_i} - v_{2,x_i}, v'_{3,x_i} - v_{3,x_i}, v'_{4,x_i} - v_{4,x_i}, v'_{5,x_i} - v_{5,x_i}, \dots\right)^T \ (i = 1, 2)$$

[00350] By the translation invariance property, we have

$$\begin{aligned} F_i(\mathbf{x}) &= U_{x_1} * N^i_{x_1} + U_{x_2} * N^i_{x_2} \\ &= \left[U_{x_1} - \left[v'_{2,x_1} - v_{2,x_1}\right]\right] * N^i_{x_1} + \left[U_{x_2} - \left[v'_{2,x_2} - v_{2,x_2}\right]\right] * N^i_{x_2} \end{aligned}$$

where $[v'_{2,x_1} - v_{2,x_1}]$ stands for a matrix whose all entries equal $v'_{2,x_1} - v_{2,x_1}$. The displacement component used to compute the internal force F_i at vertex v_3 is then

$$\begin{aligned} U_{x_k}[v_3] &= \left(v'_{3,x_k} - v_{3,x_k}\right) - \left(v'_{2,x_k} - v_{2,x_k}\right) \\ &= \left(v'_{3,x_k} - v'_{2,x_k}\right) - \left(v_{3,x_k} - v_{2,x_k}\right) \end{aligned}$$

which is the relative displacement of the vertex v_3 with respect to v_2. However, nothing prevents us from computing the relative displacement of v_3 with respect to v_3. In this case, the x_k displacement component used to compute the internal force F_i at vertex v_3 would be

$$\begin{aligned} U_{x_k}[v_3] &= \left(v'_{3,x_k} - v_{3,x_k}\right) - \left(v'_{4,x_k} - v_{4,x_k}\right) \\ &= \left(v'_{3,x_k} - v'_{4,x_k}\right) - \left(v_{3,x_k} - v_{4,x_k}\right) \end{aligned}$$

[00351] In order to take into account these facts, we use the average value of the relative displacements for all adjacent vertices to v_2. Thus

$$U_{x_k}[v_2] = \frac{1}{2}\left[\left(v'_{3,x_k} - v'_{2,x_k}\right) - \left(v_{3,x_k} - v_{2,x_k}\right) + \left(v'_{3,x_k} - v'_{4,x_k}\right) - \left(v_{3,x_k} - v_{4,x_k}\right)\right] \tag{97}$$

[00352] Reorganizing the terms of Equation 97, we have for an arbitrary vertex v_i of S

$$U_{x_k}\left[v_i\right] = \left[\frac{v_{i+1,x_k} - 2v_{i,x_k} + v_{i-1,x_k}}{2}\right] - \left[\frac{v'_{i+1,x_k} - 2v'_{i,x_k} + v'_{i-1,x_k}}{2}\right]$$

or

$$\begin{aligned} \mathrm{U}\left[v_i\right] &= \left[\frac{v_{i+1} - 2v_i + v_{i-1}}{2}\right] - \left[\frac{v'_{i+1} - 2v'_i + v'_{i-1}}{2}\right] \\ &= \frac{1}{2}\left[\frac{\partial^2 S}{\partial v^2} - \frac{\partial^2 S'}{\partial v^2}\right] \end{aligned} \qquad (98)$$

if we assume a finite difference approximation of the second derivative of S and S' with respect to their vertices.

[00353] Let us finally notice that the second derivative of the curve S in Equation 98 can be computed using Gaussian derivatives

$$\frac{\partial^2 S}{\partial v^2} = g''_\sigma * S$$

where σ controls the degree of smoothing. Such a computation of the second derivative of S allows to obtain smooth results by simulating a smoother target curve.

Experimental Results

Active Contours

[00354] The approach proposed herein has been experimented on real and synthetic images in the context of high-resolution images of road databases. For each image, we have compared our results with another method.

[00355] Figure 55(a) presents the results for our physics-based method (PB) for an aerial image while Figure 55(b) shows the results for the finite element (FEM) method (α=???, β=???). We simulated a material similar to rubber (see the Table in Figure 53) with E=150 and ν=0.45. In both images, the image force is the gradient (σ=1.5) of the bright line plausibility image

obtained using a line detector proposed in [54] with the line detection scale set to 0.8. Figures 54(a) and 54(b) show respectively the initial curve S_0 and the bright line plausibility image.

[00356] Figure 56 shows a SAR image in which the initial curves are drawn in white. The line plausibility image (σ=1.5 and line detection scale =0.8) of both bright and dark lines is shown in Figure 57. Figures 58 and 59 presents respectively the results obtained with PB (E=150 and ν=0.45) and FEM (α=??? and β=???) methods. One can notice that the PB curves are closer to the shore than the FEM especially in region of high curvature.

[00357] Figure 60 shows some initialization for the first band of a Landsat 7 image (courtesy of XXX). The bright line plausibility image (σ=1.5 and line detection scale =0.8) is shown in Figure 61. Figures 62 and 63 present the results obtained with the PB and FEM methods respectively.

[00358] We have discussed the fact that the deformations obtained using our model have a physical interpretation. We also discussed the fact that the objects modeled using the PB method have their own physical properties and the ability to recover their original shape when the external forces applied on them are removed. To illustrate this fact, Figures 64(a) and 64(b) show some initialization and the corrected curve for a synthetic image. Figures 65(a) through Figures 65(d) show the evolution of this corrected curve when the external forces are removed. Figure 65(d) presents both the final curve (in black) and the initial curve (in white). One can clearly notice that the curve has recovered its original shape but has also experienced a spatial shift.

Conclusion

[00359] A new model for active contours was presented. The proposed approach decompose the image using an image model based on algebraic topology. This model uses generic mathematical tools which can be applied to solve other problems such as linear and non-linear diffusion and

optical flow [57]. Moreover, our model works with either 2D or 3D images and can easily be extended to active surfaces and active volumes.

[00360] Our approach presents the following differences with the other methods: 1) We use both global and local quantities; 2) Our model is based upon basic laws of physics. This allows us to give a physical explanation to the deformation steps; 3) The curves and surfaces have physical behaviors such as the ability to recover their original shape once the applied forces are removed; 4) We make approximation only when the constitutive equation is involved.

[00361] Although the present invention has been described hereinabove by way of preferred embodiments thereof, it can be modified, without departing from the spirit and nature of the subject invention.

References

[00362] [1] M. Allili and D. Ziou, Extraction of topological properties of images via cubical homology, Technical Report, 2000.

[00363] [2] M. Allili and D. Ziou, Topological Feature Extraction in Binary Images, IEEE ISSPA, Malysia, Aug. 2001.

[00364] [3] M. Allili, K. Mischaikow, and A. Tannenbaum, Cubical Homology and the Topological Classification of 2D and 3D Imagery, Int. Conf. Image Processing, Greece, 2001.

[00365] [4] M. F. Auclair-Fortier, P. Poulin, D. Ziou, and M. Allili, Computational Algebraic Topology for Resolution of the Poisson Equation: Application to Computer Vision, Technical Report, 2001.

[00366] [5] R. Egli and N. F. Stewart, A framework for system specification using chains on cell complexes, Computer Aided Design 32, 447-459, 2000.

[00367] [6] P.J. Giblin, Graphs, Surfaces and Homology, Chapman and Hall; London, 1977.

[00368] [7] T.Y. Kong and A. Rosenfeld, Digital Topology: Introduction and Survey, CVGIP 48, 357-393, 1989.

[00369] [8] V.A. Kovalvesky, Finite Topology to Image Analysis, Comp. Vision, and Image Processing 46, 141-161, 1989.

[00370] [9] W.S. Massey, A Basic Course in Algebraic Topology, Springer-Verlag, New York, 1991.

[00371] [10] J.R. Munkres, Elements of Algebraic Topology, Addison-Wesley, 1984.

[00372] [11] R.S. Palmer and V. Shapiro, Chain Models of Physical Behavior for Engineering Analysis and Design, Research in Engineering Design 5, 161-184, 1993.

[00373] [12] P. Poulin, D. Ziou, and M.F. Auclair-Fortier, Computational Algebraic Topology for Deformable Objects, Technical Report, 2001.

[00374] [13] A.S. Schwarz, Topology for Physicists, Springer-Verlag, 1994.

[00375] [14] E. Tonti, On the Formal Structure of Physical Theories, Technical Report Istituto Di Mathematica Del Politechnico Di Milano, Milan, 1975.

[00376] [15] D. Ziou and M. Allili, Computation of the Euler Number in Binary Images without Skeletonization via Cubical Complex, Technical Report, 2001.

[00377] [16] M. Allili and D. Ziou. Extraction of Topological Properties of Images via Cubical Homology. Technical Report CDSNS 2000-365, Georgia Institute of Technology, June 2000.

[00378] [17] L. Alvarez, R. Deriche, and F. Santana. Recursivity and PDE's in Image Processing. In Proceedings 15th International Conference on Pattern Recognition, volume I, pages 242--248, September 2000.

[00379] [18] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Systems and Experiment Performance of Optical Flow Techniques. International Journal of Computer Vision, 12(1):43--77, 1994.

[00380] [19] S.S. Beauchemin and J.L. Barron. The Computation of Optical Flow. ACM Computing Survey, 27(3), 1995.

[00381] [20] A. J. Chapman. Fundamentals of Heat Transfer. Macmillan Publishing Company, 1987.

[00382] [21] A.K. Chhabra and T.A.Grogan. On Poisson Solvers and Semi-Direct Methods for Computing Area Based Optical Flow. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16:1133--1138, 1994.

[00383] [22] H. Chidiac and D. Ziou. Formation d'images optiques. Technical Report 226, Département de mathématiques et d'informatique, Université de Sherbrooke, November 1998.

[00384] [23] L. D. Cohen and I. Cohen. Finite Element Methods for Active Contour Models and Balloons for 2D and 3D Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15, November 1993.

[00385] [24] L.D. Cohen. On active contour models and balloons. Computer Vision, Graphics and Image Processing, 53(2):211--218, March 1991.

[00386] [25] R. Deriche and O. Faugeras. Les EDP en traitement des images et vision par ordinateur. Technical Report 2697, INRIA, November 1995.

[00387] [26] C. H. Edwards and D. E. Penney. Calculus with Analytic Geometry. Prentice Hall, 1998.

[00388] [27] H. U. Fuchs. The Dynamics of Heat. Springer-Verlag, 1996.

[00389] [28] D. Halliday, R. Resnick, and J. Walker. Fundamentals of Physics. John Willey and Sons, 1997.

[00390] [29] B. K. P. Horn and B. Schunck. Determining Optical Flow. Artificial Intelligence, 17:185--203, 1981.

[00391] [30] J. Kervorkian. Partial Differential Equations: Analytical Solution Techniques, chapter 2, pages 48--116. Mathematics Series. Chapman and Hall, 1990.

[00392] [31] S.H. Lai and B.C. Vermuri. An O(n) Iterative Solution to the Poisson Equation in Low-level Vision Problems. Technical Report TR93-035, University of Florida, Computer and Information Sciences Department, 1993.

[00393] [32] J. Li and A. O. Hero. A Spectral Method for Solving Elliptic Equations for Surface Reconstruction and 3D Active Contours. Proceedings of IEEE International Conference on Image Processing, Thessaloniki, Greece, October 2001.

[00394] [33] G. T. Mase and G. E. Mase. Continuum Mechanics for Engineers. CRC Press, 1999.

[00395] [34] C. Mattiussi. An Analysis of Finite Volume, Finite Element, and Finite Difference Methods Using Some Concepts from Algebraic Topology. Journal of Computational Physics, 133:289--309, 1997.

[00396] [35] J. Monteil and A. Beghdadi. A New Interpretation and Improvement of the Nonlinear Anisotropic Diffusion for Image Enhancement. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(9):940--946, September 1999.

[00397] [36] P. Poulin, M.-F. Auclair-Fortier, D. Ziou and M. Allili. A Physics Based Model for the Deformation of Curves: A Computational Algebraic Topology Approach. Technical Report 270, Département de mathématiques et d'informatique, Université de Sherbrooke, 2001.

[00398] [37] S.V. Patankar. Numerical Heat Transfer and Fluid Flow. Computational Methods in Mechanics and Thermal Sciences. McGraw-Hill Book Company, 1980.

[00399] [38] P. Perona and J. Malik. Scale-Space and Edge Detection Using Anisotropic Diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(7):629--639, July 1990.

[00400] [39] T. Symchony, R. Chellappa, and M. Shao. Direct Analytical Methods for Solving Poisson Equations in Computer Vision Problems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(5):435--446, May 1990.

[00401] [40] D. Terzopoulos. Image Analysis Using Multigrid Relaxation Methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8:129--129, 1986.

[00402] [41] G. B. Thomas and R. L. Finney. Calculus and Analytic Geometry. Addison-Wesley Publishing Company, 1988.

[00403] [42] E. Tonti. On the Formal Structure of Physical Theories. Technical report, Istituto Di Mathematica Del Polithechnico Di Milano, 1975.

[00404] [43] J. Weickert. A Review of Nonlinear Diffusion Filtering. Lecture Notes in Computer Science, 1252:3--28, 1997.

[00405] [44] E. Zauderer. Partial Differential Equations of Applied Mathematics. Pure and Applied Mathematics. John Willey and Sons, New York, US, 1983.

[00406] [45] M. Allili and D. Ziou. Extraction of Topological Properties of Images via Cubical Homology. Technical Report CDSNS 2000-365, Georgia Institute of Technology, June 2000.

[00407] [46] M.-F. Auclair-Fortier, D. Ziou, C. Armenakis, and S. Wang. Automated Correction and Updating of Road Databases from High-Resolution Imagery. Canadian Journal of Remote Sensing, 27:76--89, 2001.

[00408] [47] M.F. Auclair-Fortier, P. Poulin, D. Ziou, and M. Allili. Physics Based Resolution of Smoothing and Optical Flow: A Computational Algebraic Topology Approach. Technical Report 269, Département de mathématiques et d'informatique, Université de Sherbrooke, 2001.

[00409] [48] K.J. Bathes. Finite element procedures. Prentice Hall, 1996.

[00410] [49] A.P. Boresi. Elasticity in Engineering Mechanics. Prentice Hall, 1965.

[00411] [50] A.P. Boresi, R.J. Schmidt, and O.M. Sidebottom. Advanced Mechanics of Materials, Fifth edition. John Willey and Sons, 1993.

[00412] [51] M. Bro-Nielsen. Surgery simulation using fast finite elements. In Proc. Visualization in Biomedical Computing (VBC'96), volume 1131, pages 529--534, Hamburg, Germany, September 1996. Springer Lecture Notes in Computer Science.

[00413] [52] E.F. Byars and R.D. Snyder. Mechanics of Deformable Bodies. Intext Educational Publishers, 1975.

[00414] [53] S. Cotin and N. Ayache. A Hybrid Elastic Model Allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation. In Proc. of CAS99, pages 70--81, May 1999.

[00415] [54] F. Deschênes, D. Ziou, and M.-F. Auclair-Fortier. Detection of Lines, Line Junctions and Line Terminations. Technical Report 259, Département de mathématiques et d'informatique, Université de Sherbrooke, 2000. Submitted in ISPRS Journal of Photogrammetry and Remote Sensing, 2000.

[00416] [55] C. H. Edwards and D. E. Penney. Calculus with Analytic Geometry. Prentice Hall, 1998.

[00417] [56] K.M. Entwistle. Basic Principles of the Finite Element Method. IOM Communications Ltd, 1999.

[00418] [57] S.F.F. Gibson and B. Mirtich. A Survey of Deformable Modeling in Computer Graphics. Technical report, Mitsubishi Electric Research Laboratory, 1997.

[00419] [58] R.C. Juvinall. Engineering considerations of Stress, Strain ans Strength. McGraw-Hill, 1967.

[00420] [59] M. Kass, A. Witkin, and D. Terzopoulos. Snakes : Active Contour Models. The International Journal of Computer Vision, 1(4):321–331, 1988.

[00421] [60] G. T. Mase and G. E. Mase. Continuum Mechanics for Engineers. CRC Press, 1999.

[00422] [61] C. Mattiussi. An Analysis of Finite Volume, Finite Element, and Finite Difference Methods Using Some Concepts from Algebraic Topology. Journal of Computational Physics, 133:289–309, 1997.

[00423] [62] J. Montagnat and H. Delingette. Volumetric Medical Images Segmentation using Shape Constrained Deformable Models. Lecture Notes in Computer Science, 1205:13–22, 1997.

[00424] [63] J. Montagnat, H. Delingette, N. Scapel, and N. Ayache. Representation, shape, topology and evolution of deformable surfaces. Application to 3D medical imaging segmentation. Technical Report 3954, INRIA, 2000.

[00425] [64] K.W. Morton and D.F. Mayers. Numerical Solution of Partial Differential Equations. Cambridge University Press, 1994.

[00426] [65] B.V. Muvdi and J.W. McNabb. Engineering Mechanics of Materials. Macmillan Publishing Co, 1980.

[00427] [66] N.O. Myklestad. Statics of deformable bodies. MacMillan Company, 1966.

[00428] [67] R. S. Palmer and V. Shapiro. Chain Models of Physical Behavior for Engineering Analysis and Design. Technical Report TR93-1375, Cornell University, Computer Science Department, August 1993.

[00429] [68] N. Peterfreund. Robust Tracking of Position and Velocity With Kalman Snakes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(6):564--569, 1999.

[00430] [69] W.D. Pilkey and W. Wunderlich. Mechanics of structures: variational and computational methods. CRC Press, 1994.

[00431] [70] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. Numerical Recipes in C. Cambridge University Press, 1992.

[00432] [71] G. B. Thomas and R. L. Finney. Calculus and Analytic Geometry. Addison-Wesley Publishing Company, 1988.

[00433] [72] E. Tonti. On the Formal Structure of Physical Theories. Technical report, Istituto Di Mathematica Del Polithechnico Di Milano, 1975.

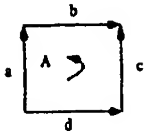


Figure 1

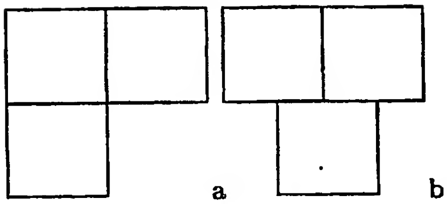


Figure 2

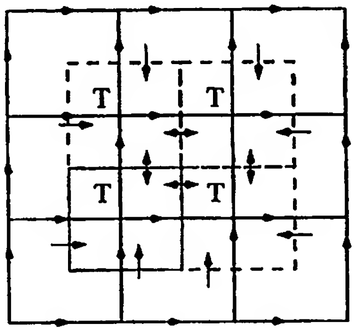


Figure 3

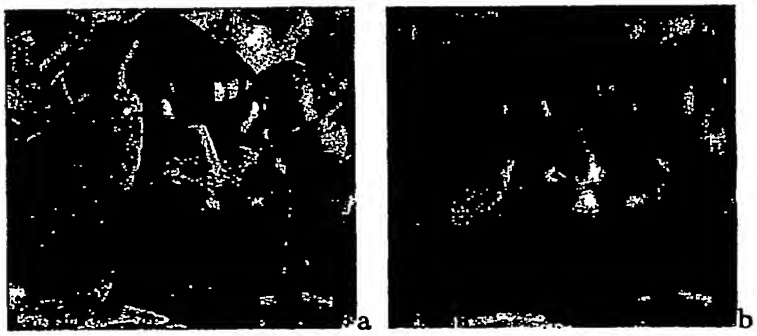


Figure 4

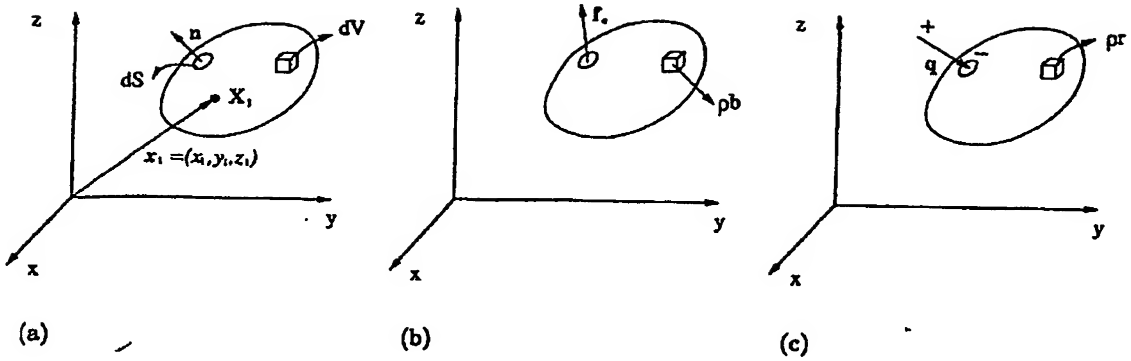


Figure 5

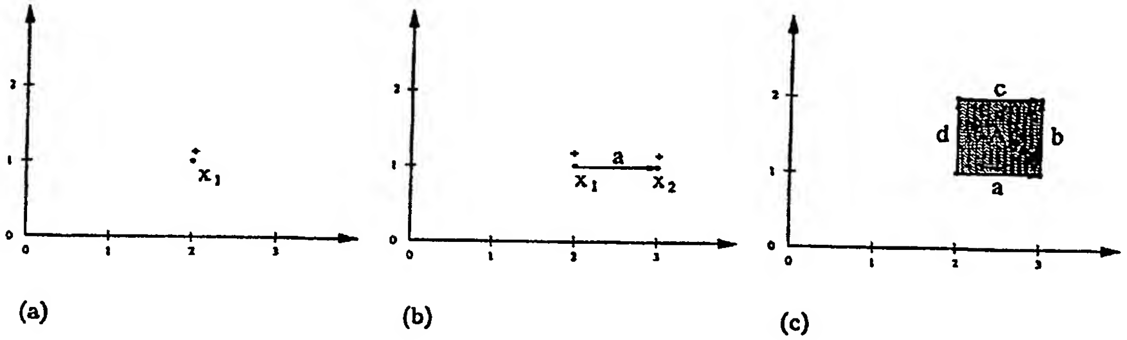


Figure 6

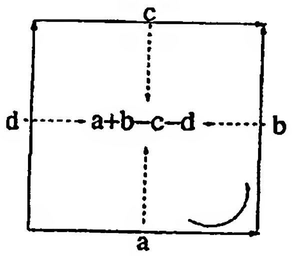


Figure 7

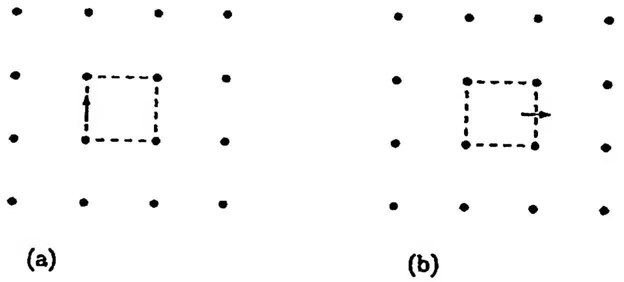


Figure 8

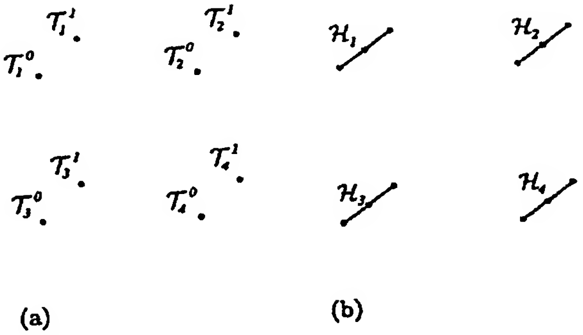


Figure 9

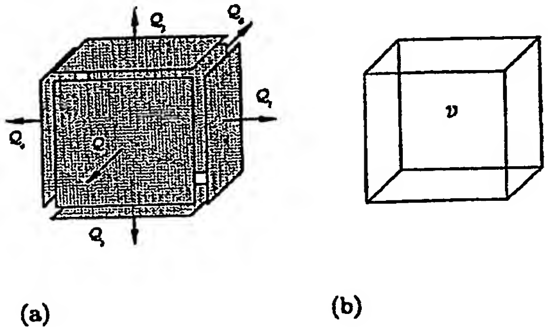


Figure 10

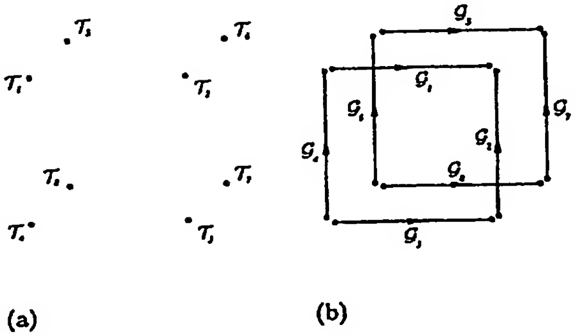


Figure 11

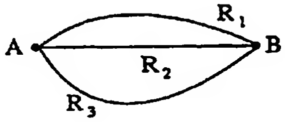


Figure 12

$$\begin{tikzcd}
 & \tilde{h} \arrow[dr] & \\
\mathcal{H} \arrow[ur] \arrow[rr] & & \Gamma \arrow[d] \\
<\mathcal{H},\gamma_H>=<\mathcal{T},\partial\gamma_H> \arrow[u] & & \mathcal{E} \arrow[d] \\
\mathcal{T} \arrow[u] \arrow[d] & & <\mathcal{E},\gamma_E>=<\mathcal{D},\gamma_E>+<\mathcal{S},\gamma_E> \\
<\mathcal{G},\gamma_G>=<\mathcal{T},\partial\gamma_G> \arrow[d] & & <\mathcal{D},\gamma_E>=<\mathcal{Q},\partial\gamma_E> \arrow[u] \\
\mathcal{G} \arrow[r] \arrow[d] & \Lambda \arrow[r] & \mathcal{Q} \arrow[u] \\
\textit{local approximation} \arrow[d] & & \int\tilde{\mathbf{q}}\cdot\mathbf{n} \arrow[u] \arrow[ul] \\
\tilde{\mathbf{g}} \arrow[r] & \overset{\textit{local equation}}{\tilde{\mathbf{q}}=\lambda\tilde{\mathbf{g}}} \arrow[r] & \tilde{\mathbf{q}} \arrow[u]
\end{tikzcd}$$

Figure 13

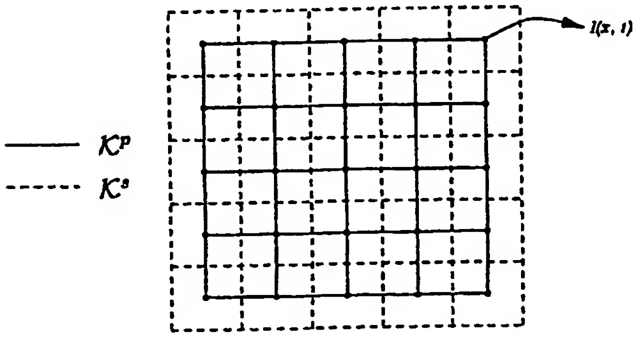


Figure 14

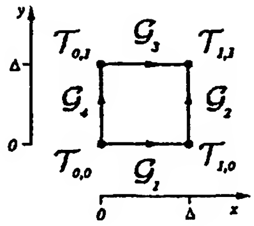
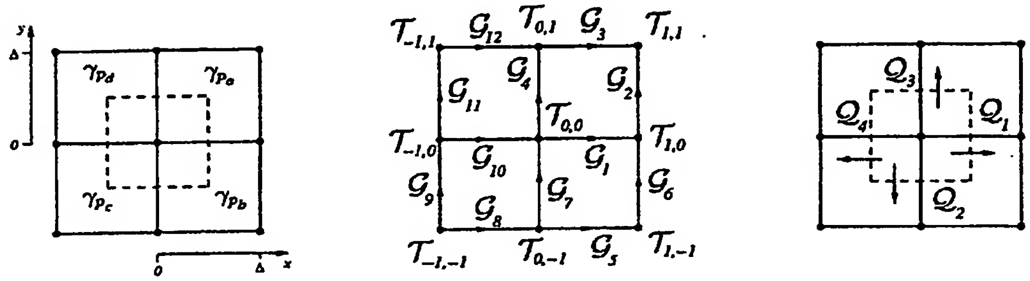


Figure 15



(a)

(b)

(c)

Figure 16

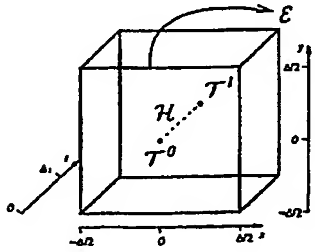


Figure 17

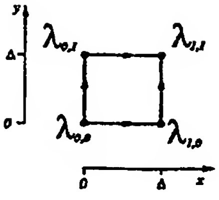
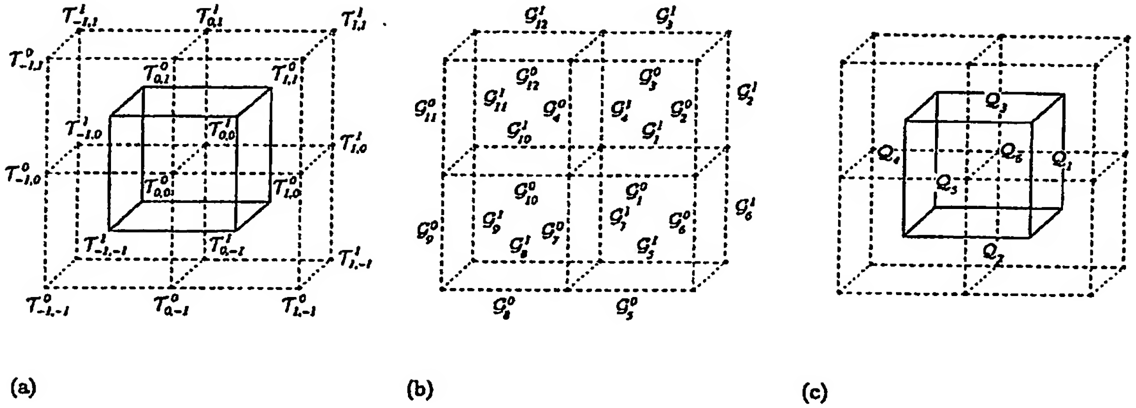


Figure 18



(a)

(b)

(c)

Figure 19

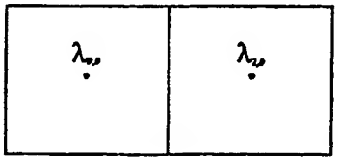


Figure 20

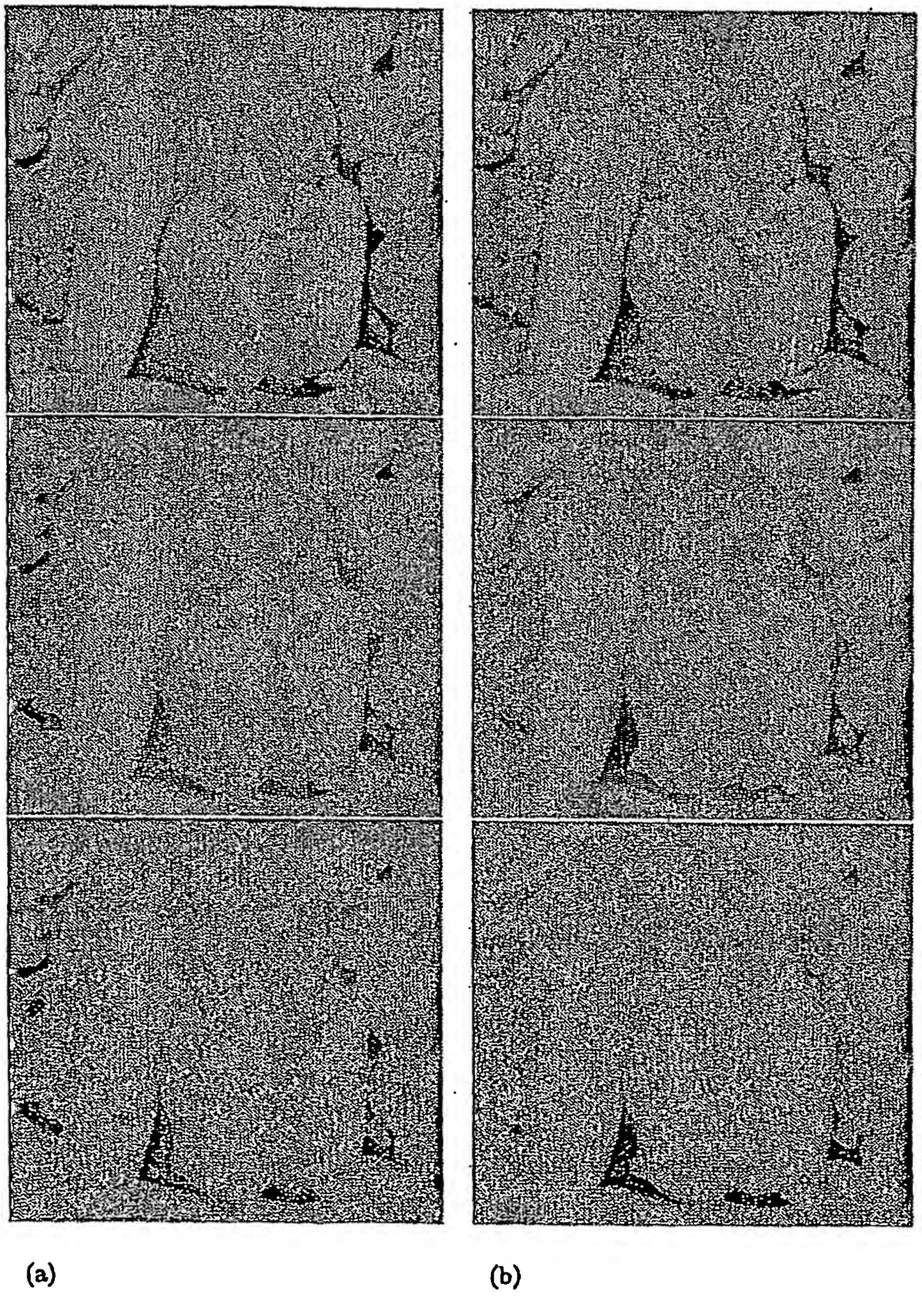


Figure 21

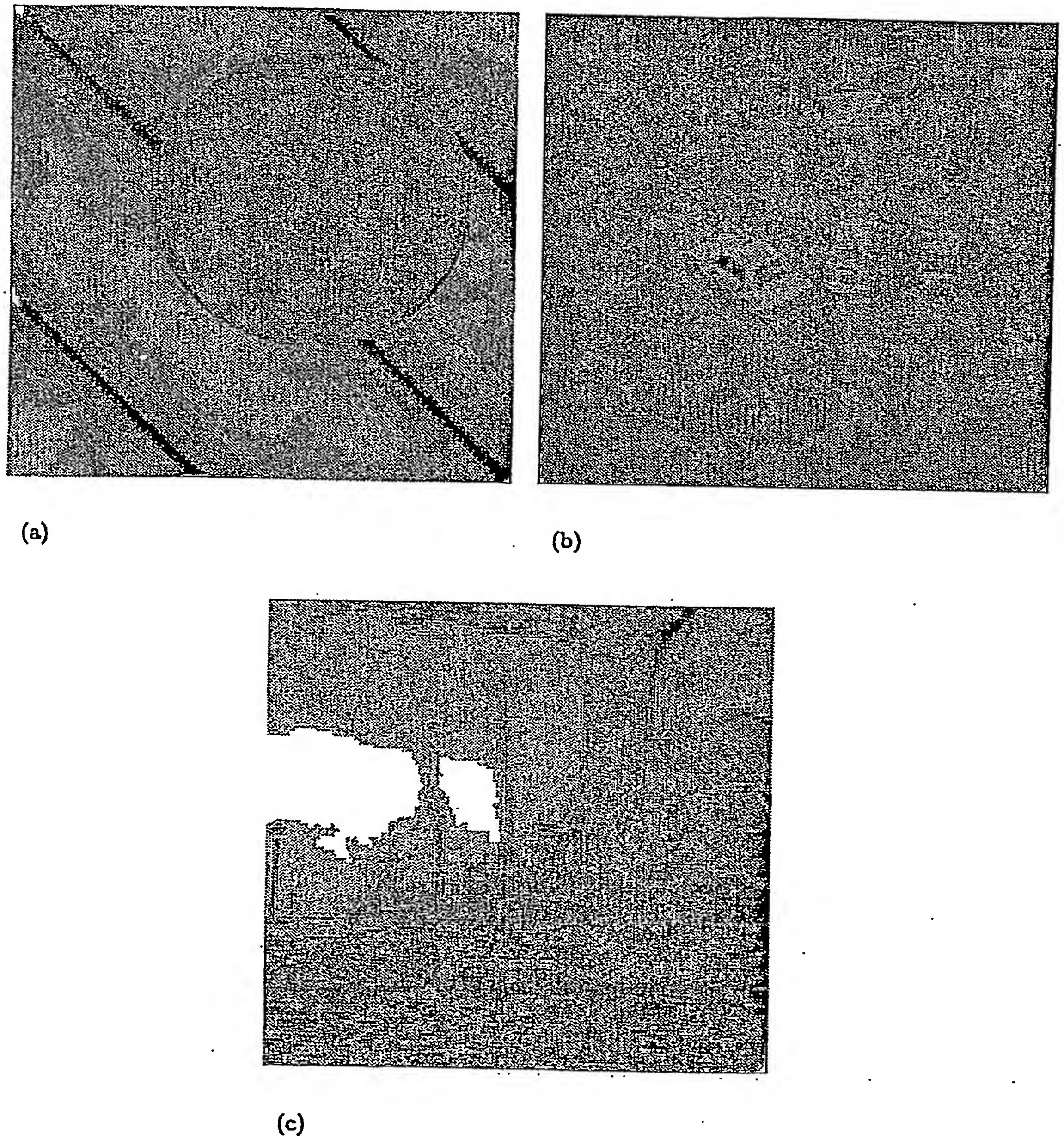
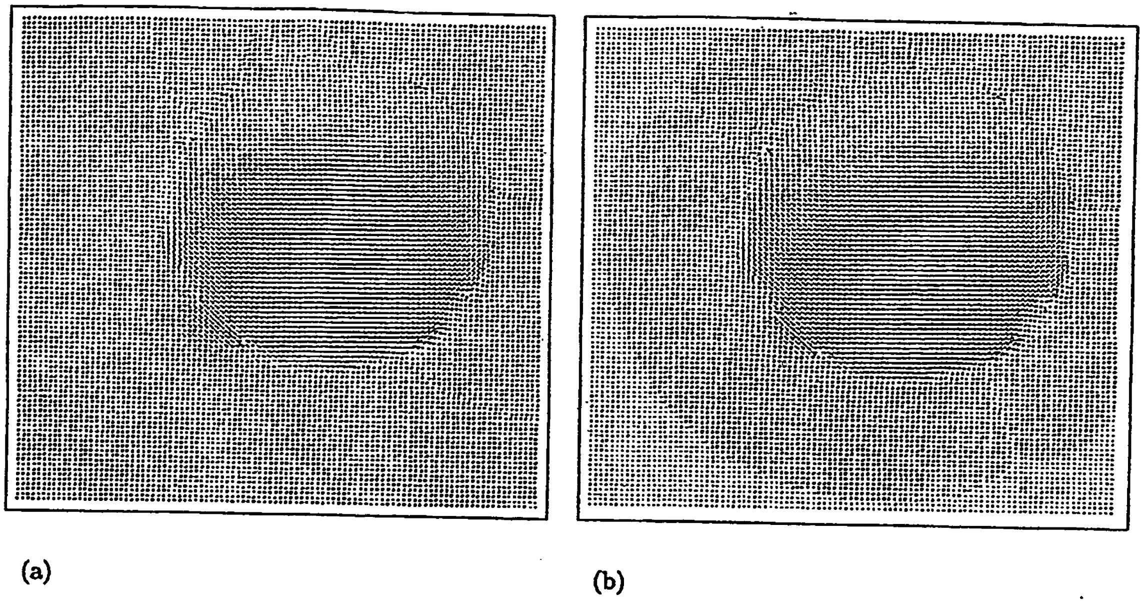
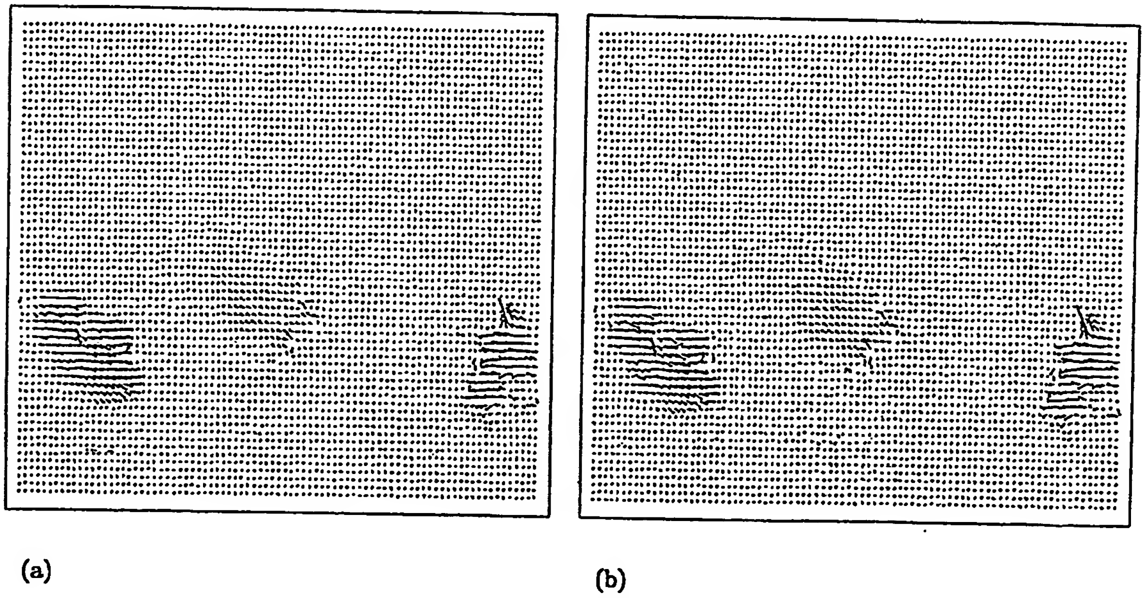


Figure 22



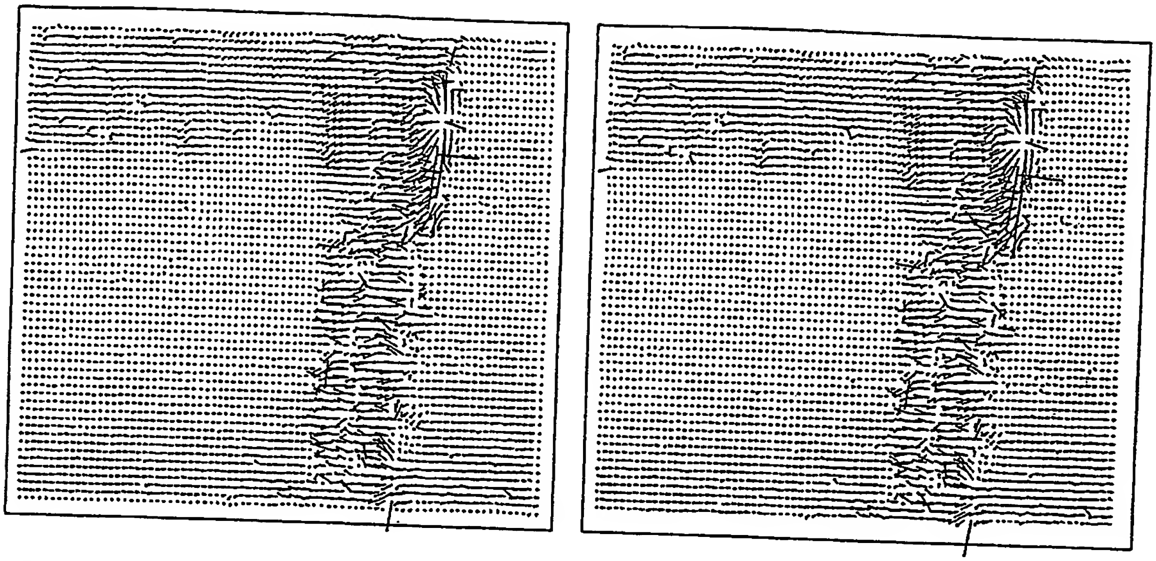
(a) (b)

Figure 23



(a) (b)

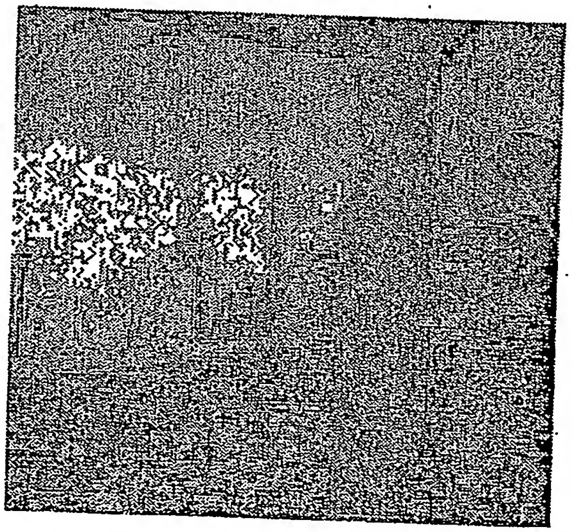
Figure 24



(a)

(b)

Figure 25



(a)

Figure 26

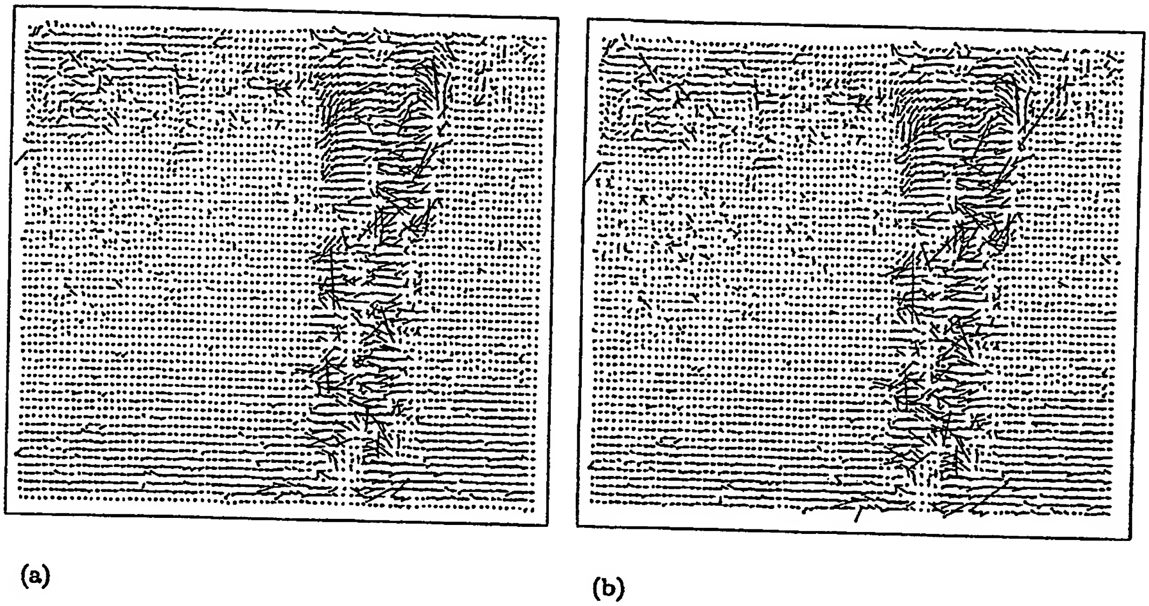


Figure 27

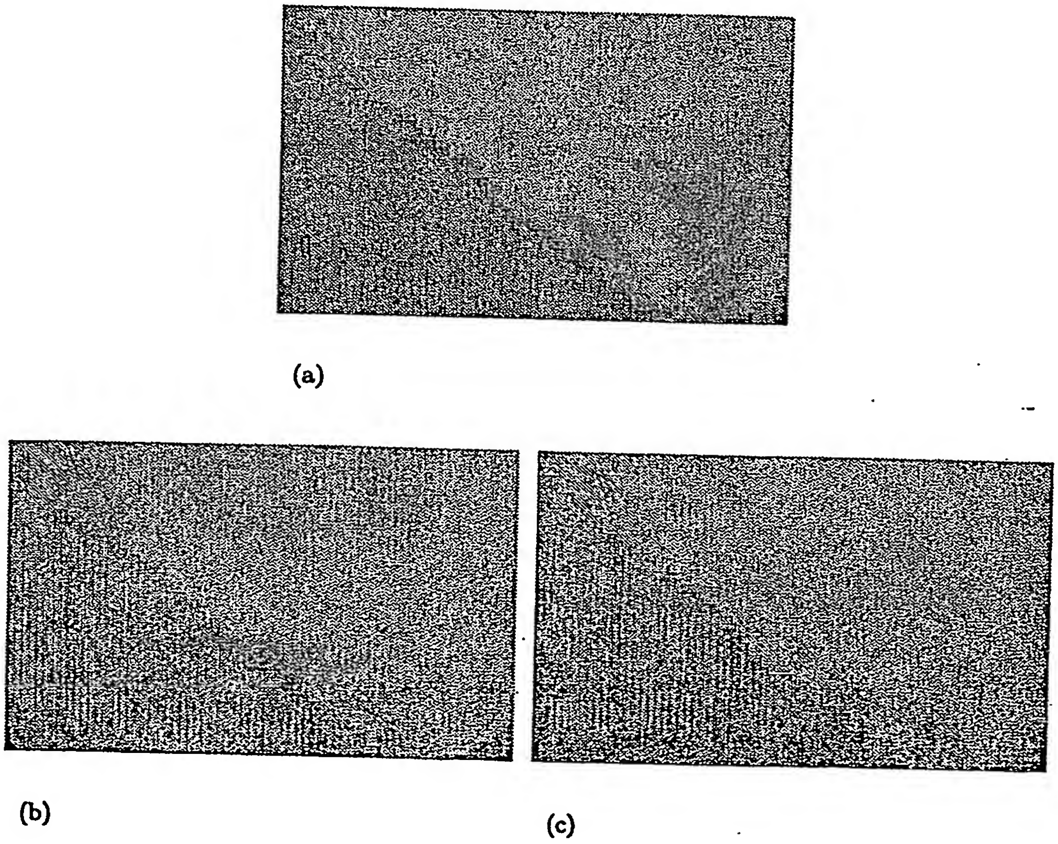
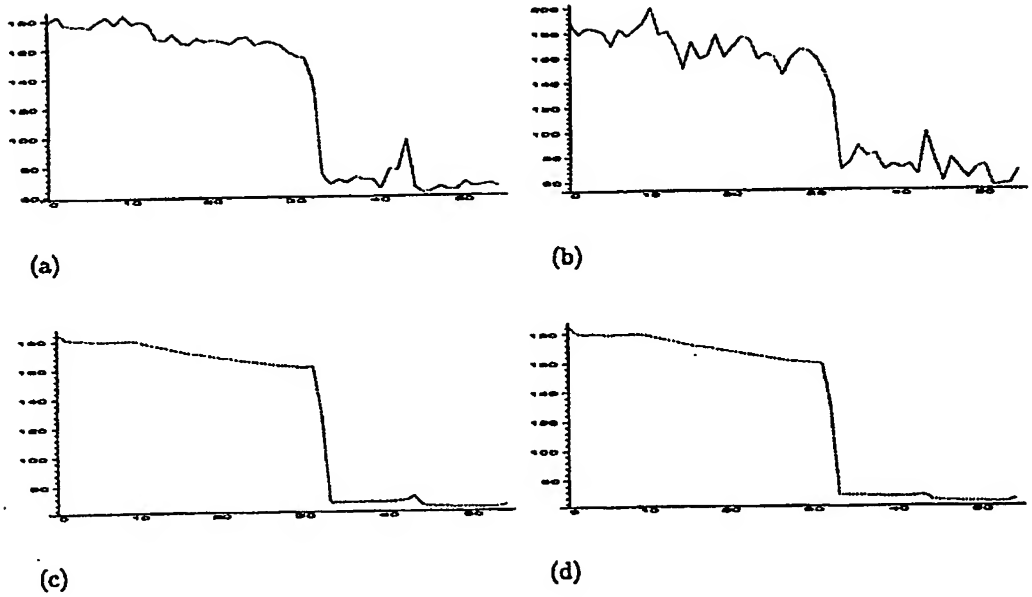


Figure 28



(a)

(b)

(c)

(d)

Figure 29

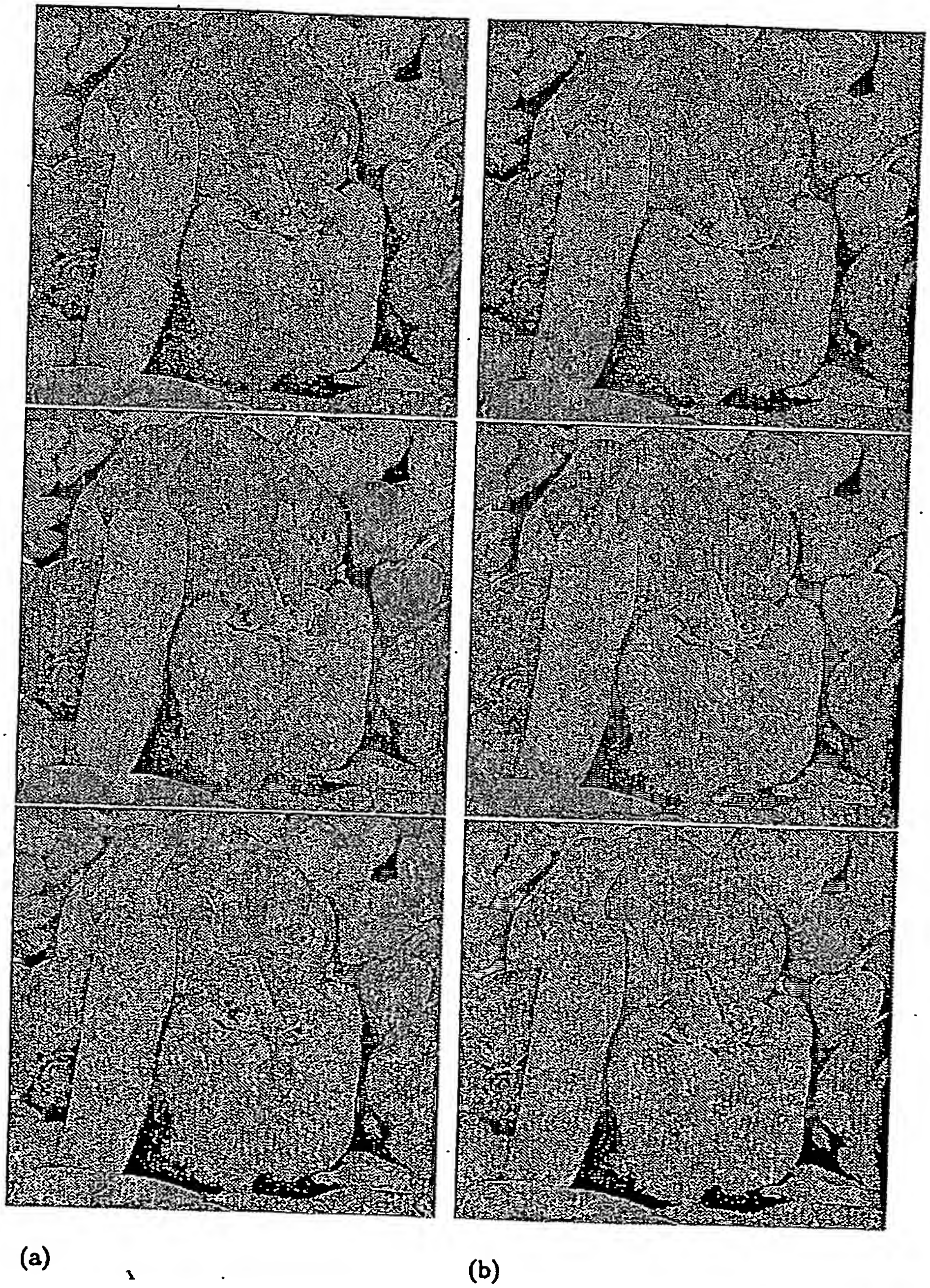


Figure 30

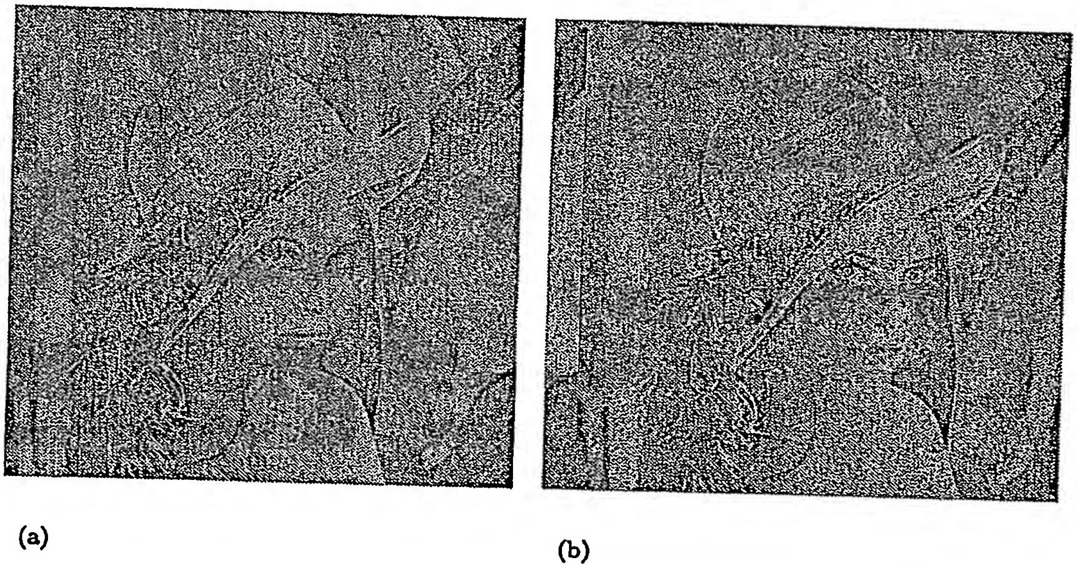


Figure 31

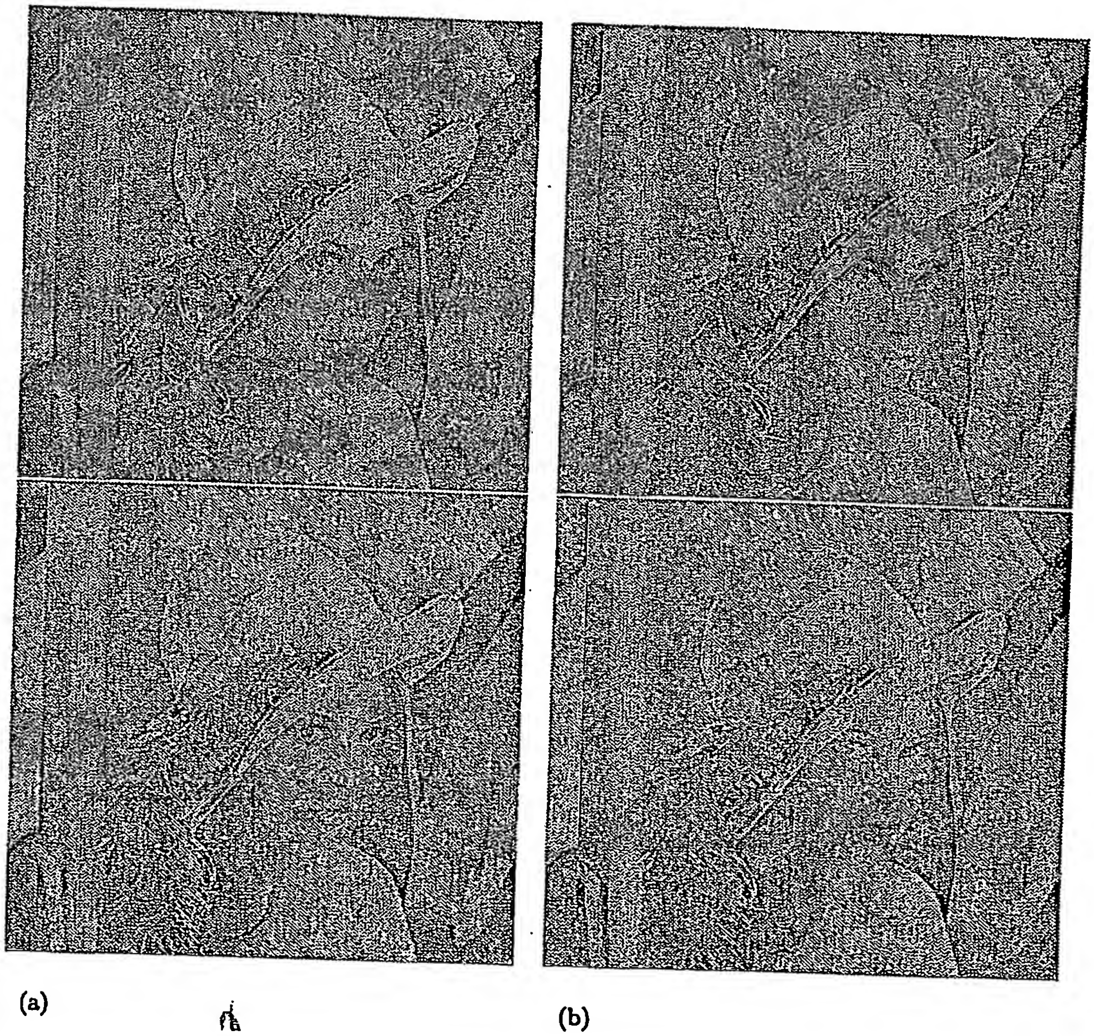
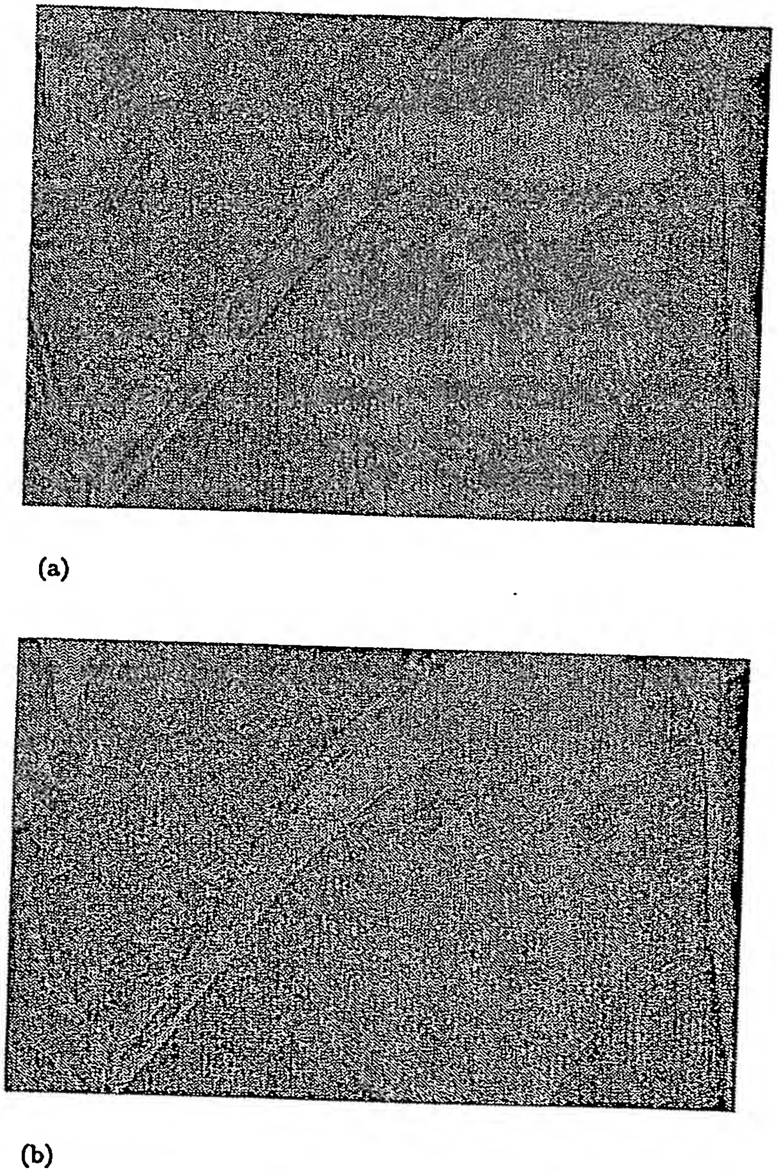


Figure 32



(a)

(b)

Figure 33

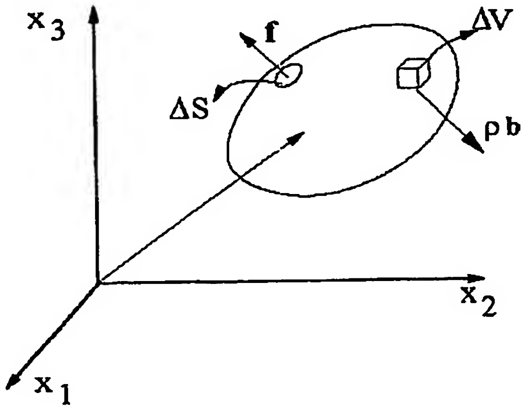


Figure 34

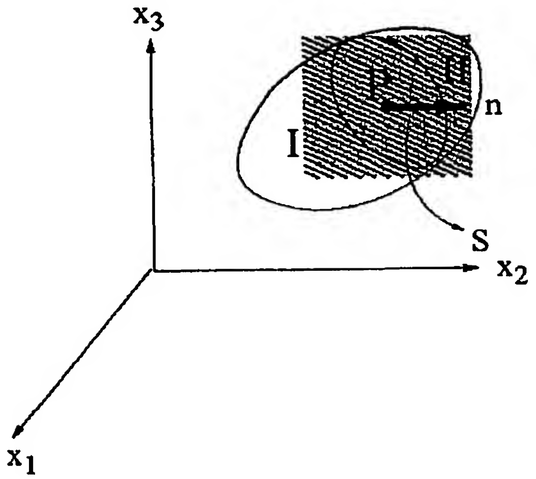


Figure 35

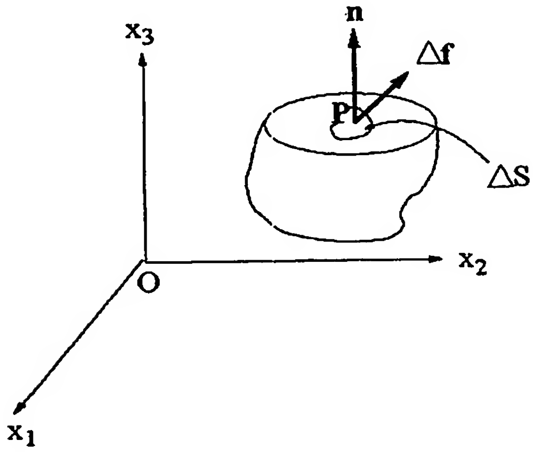
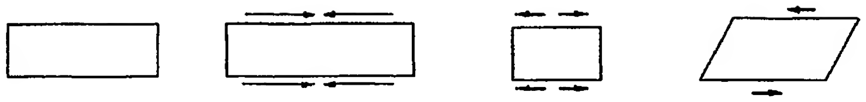


Figure 36



(a) (b) (c) (d)

Figure 37

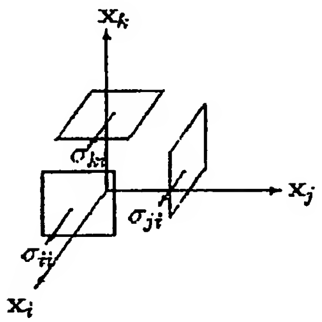
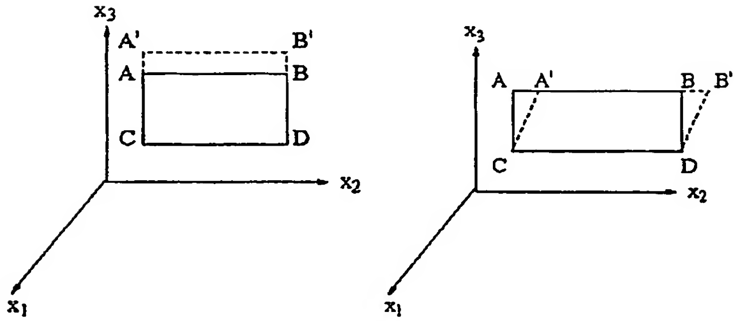


Figure 38



(a) (b)

Figure 39

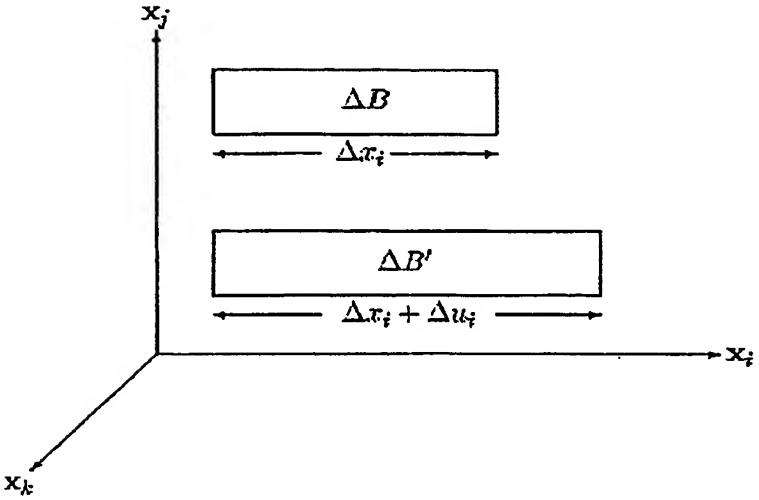


Figure 40

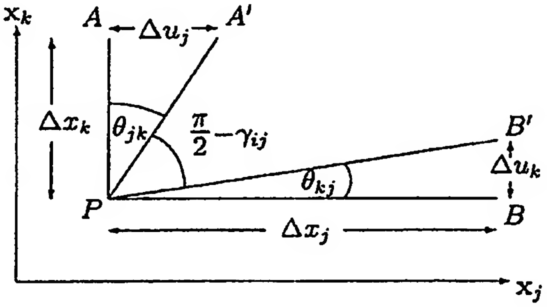


Figure 41

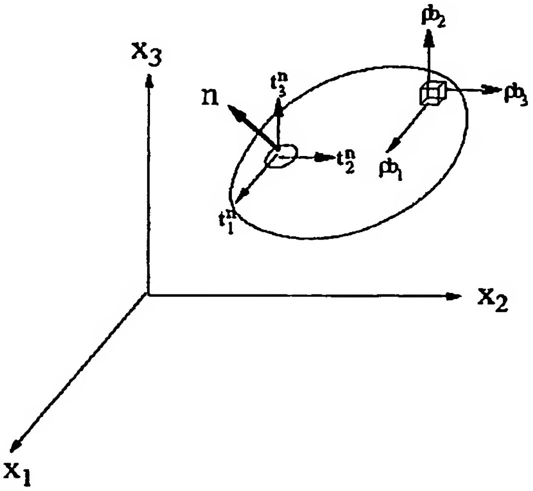


Figure 42

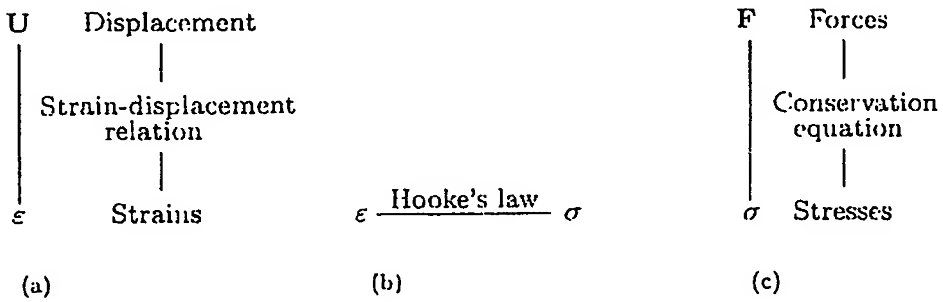


Figure 43

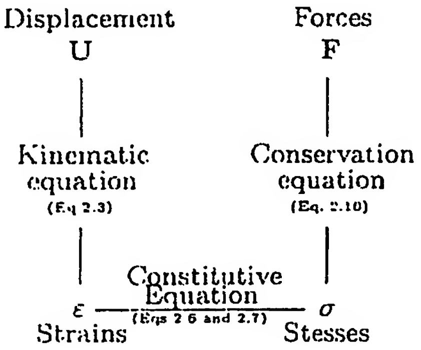


Figure 44

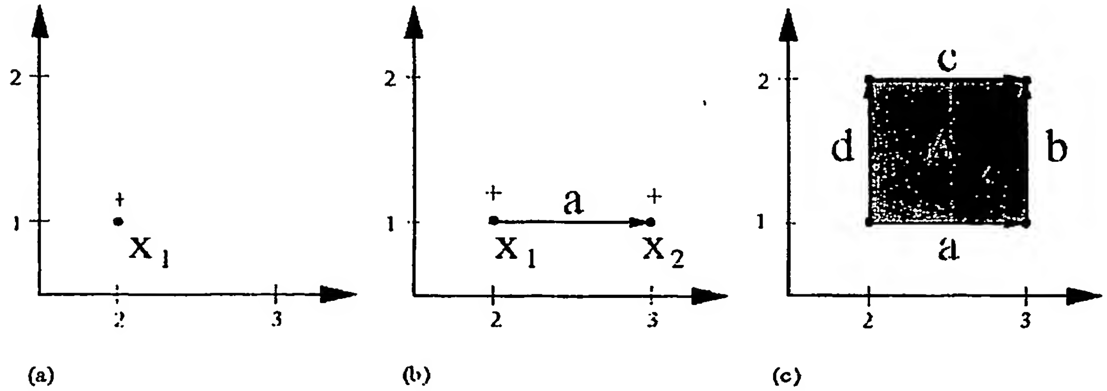


Figure 45

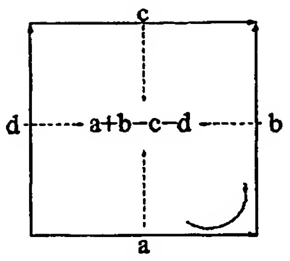


Figure 46

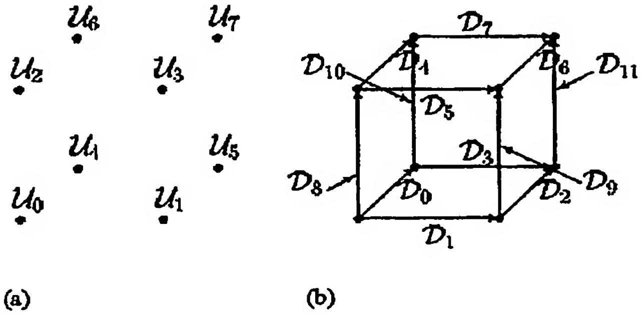


Figure 47

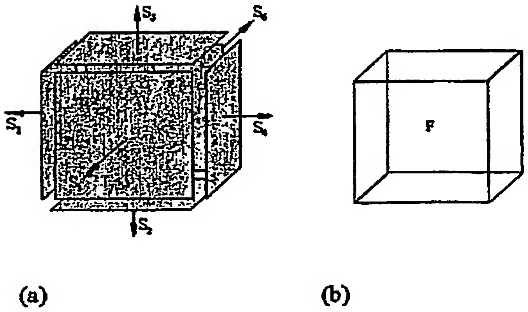


Figure 48

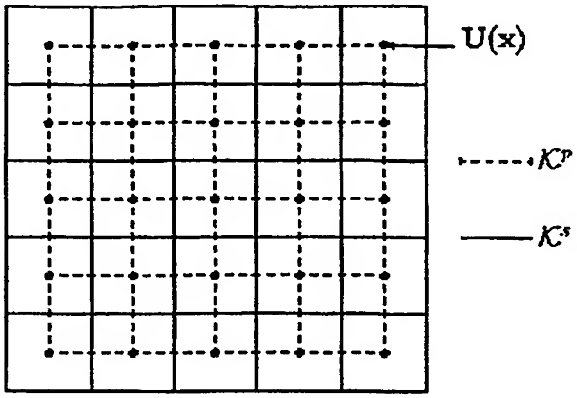


Figure 49

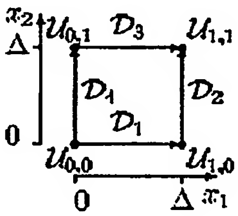


Figure 50

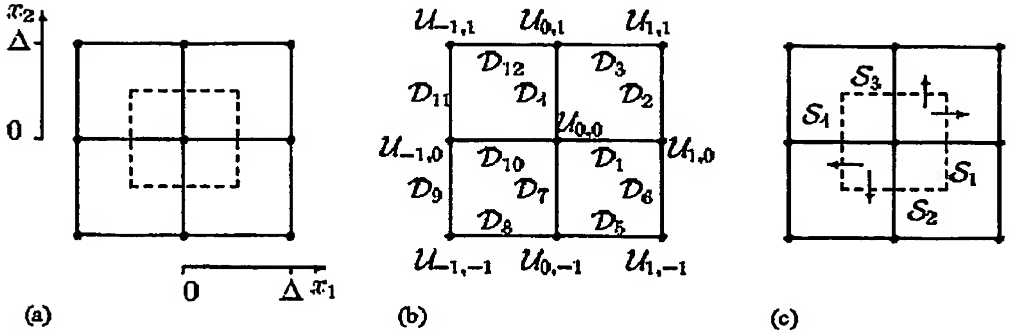


Figure 51

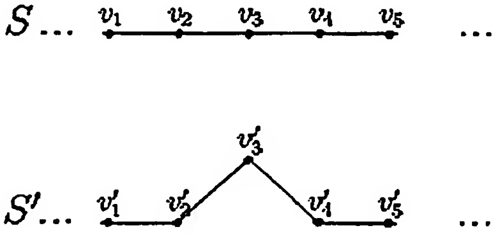


Figure 52

Material	Young's modulus of elasticity ($\times 10^6$ psi)	Poisson's ratio
Aluminium	10	0.33
Iron	29	0.28
Magnesium	6.5	0.33
Nickel	30	0.31
Soft rubber	0.00015	0.49

Figure 53

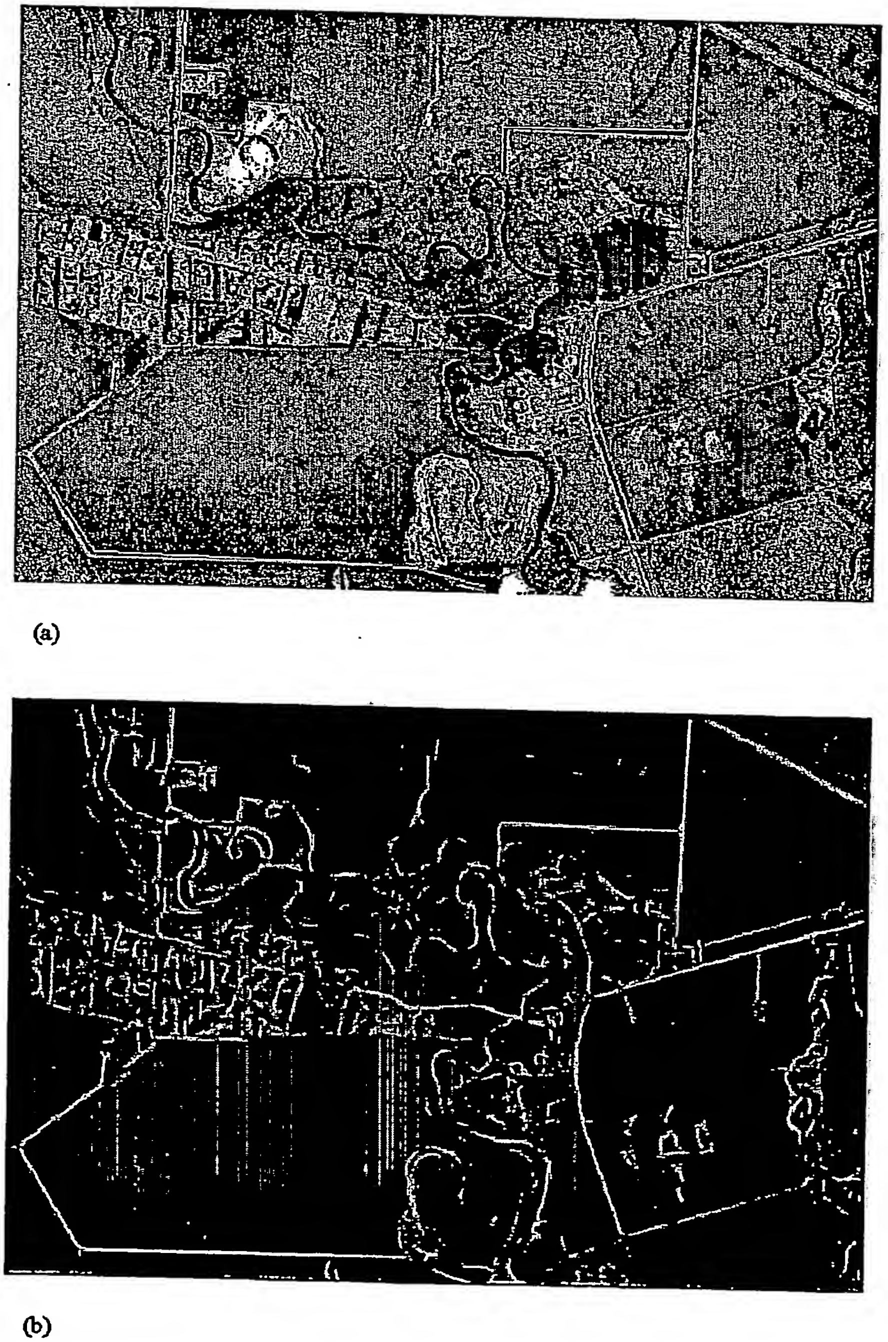
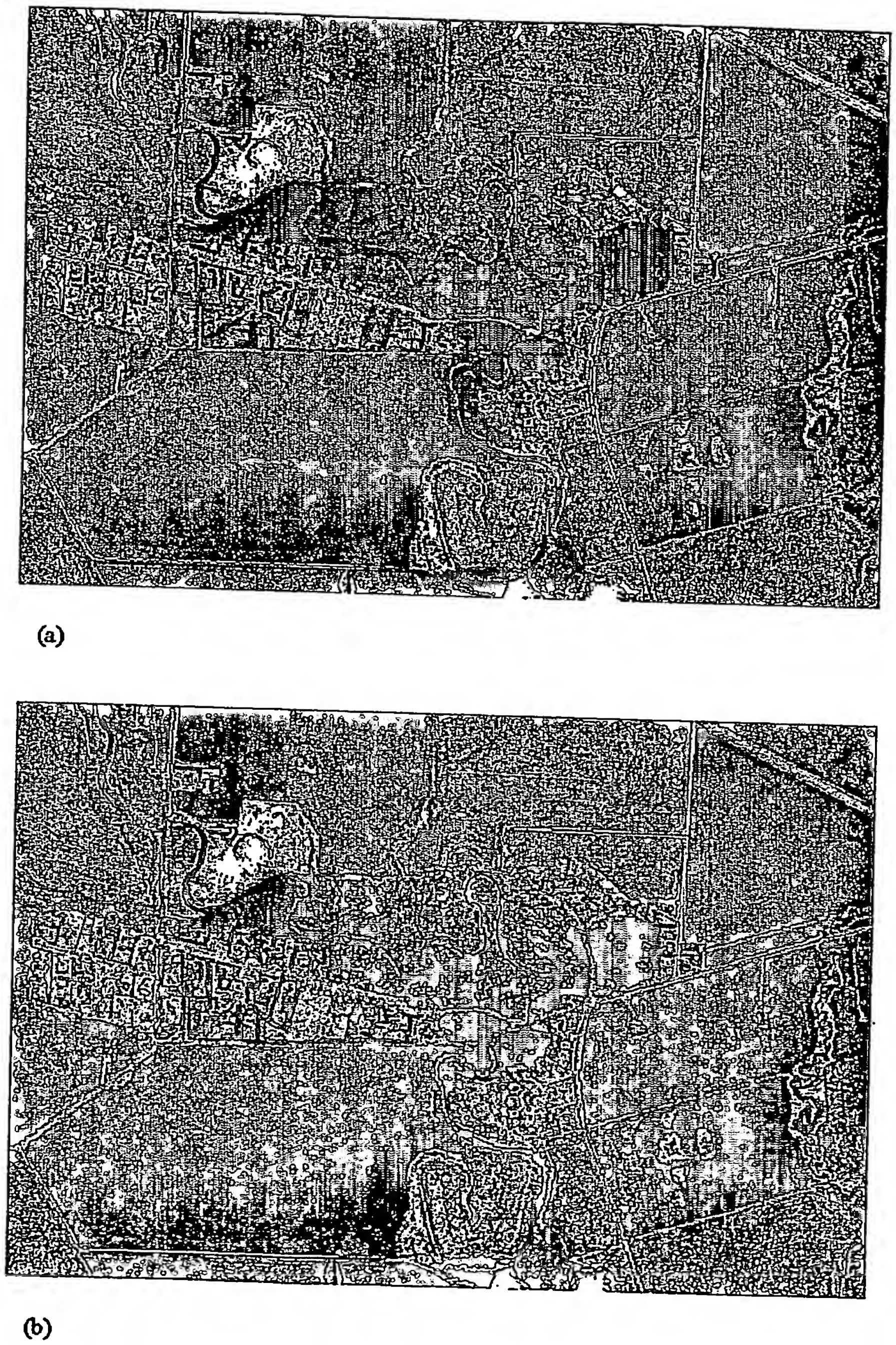


Figure 54



(a)

(b)

Figure 55

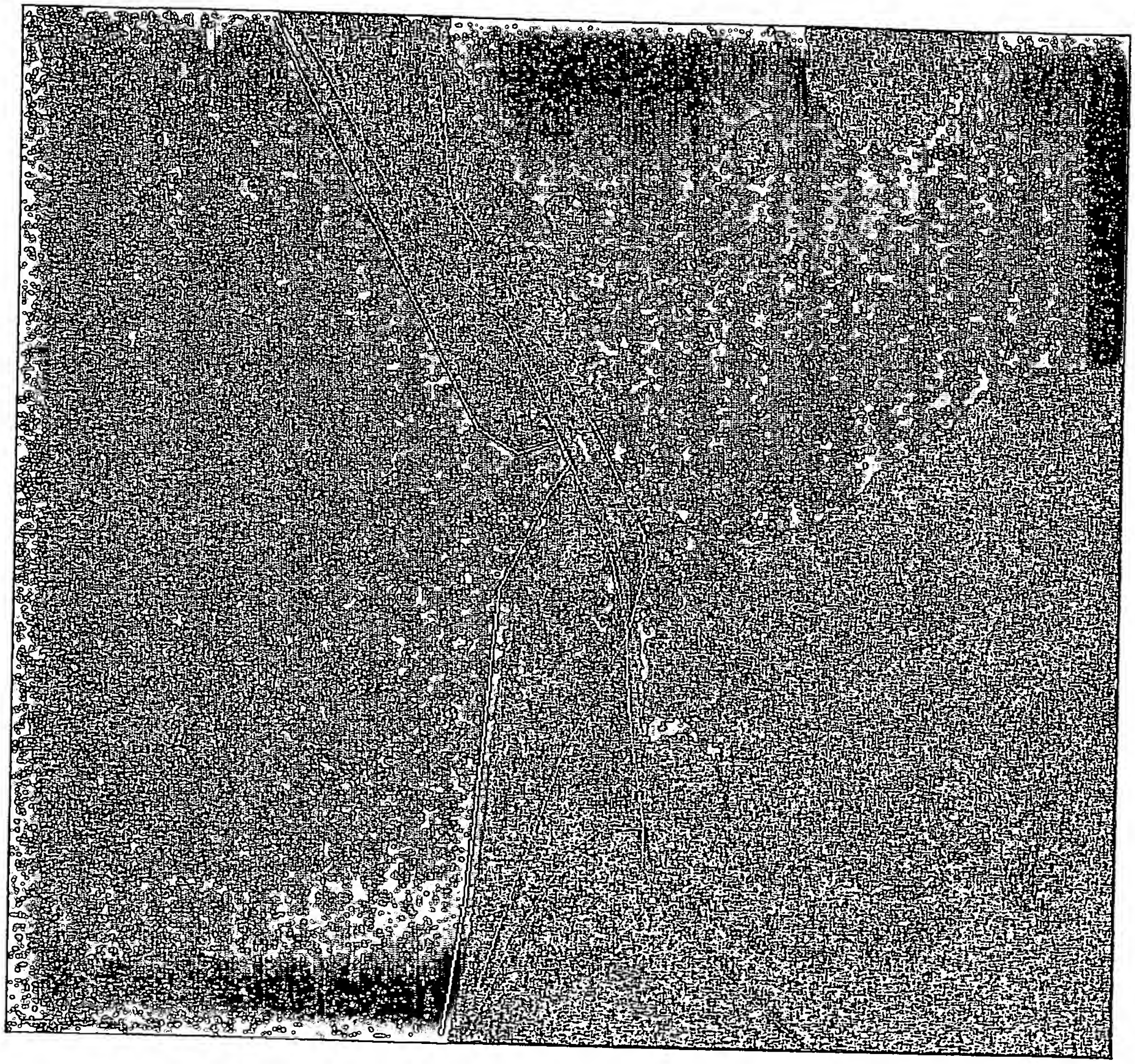


Figure 56

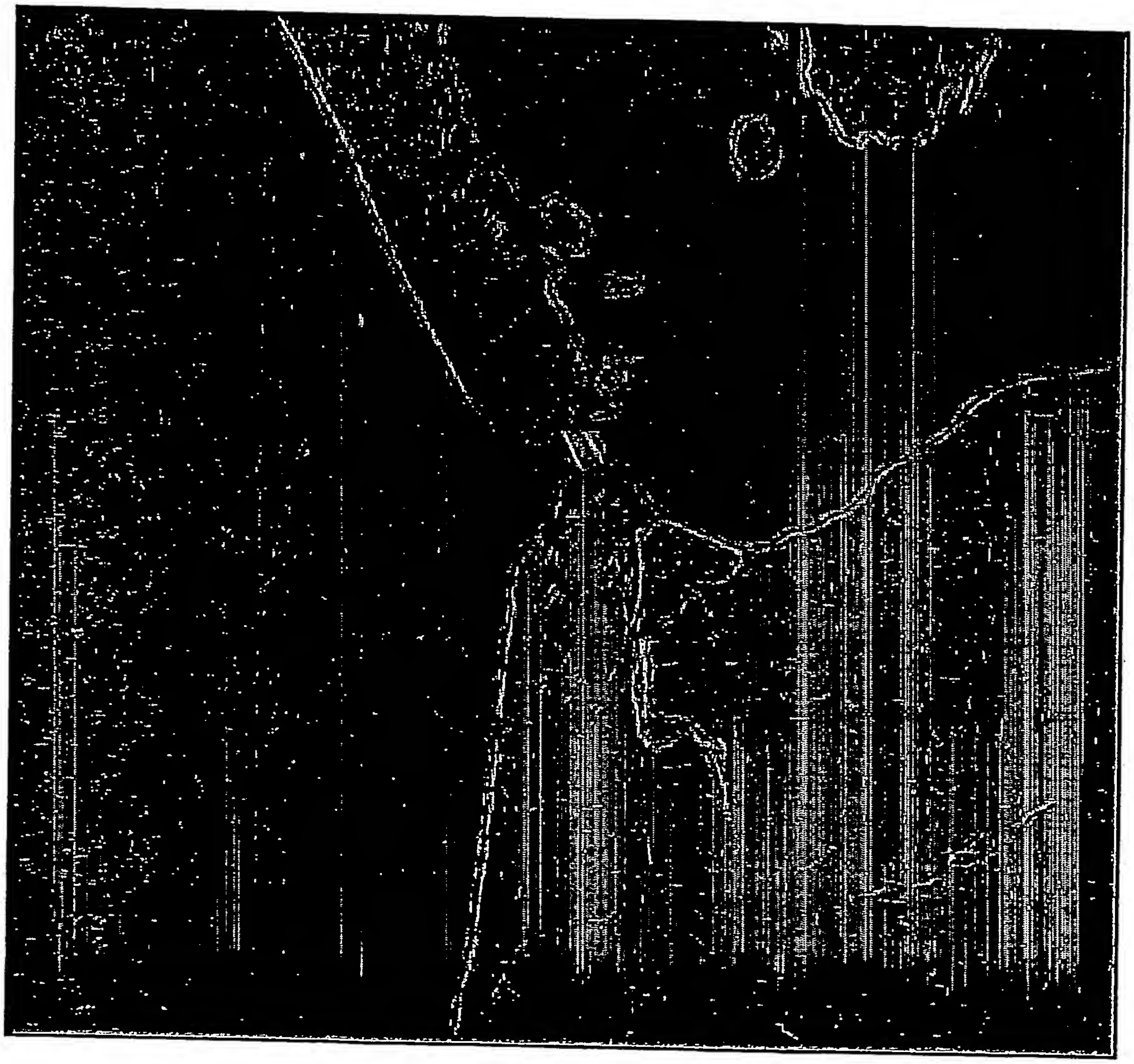


Figure 57

Figure 58

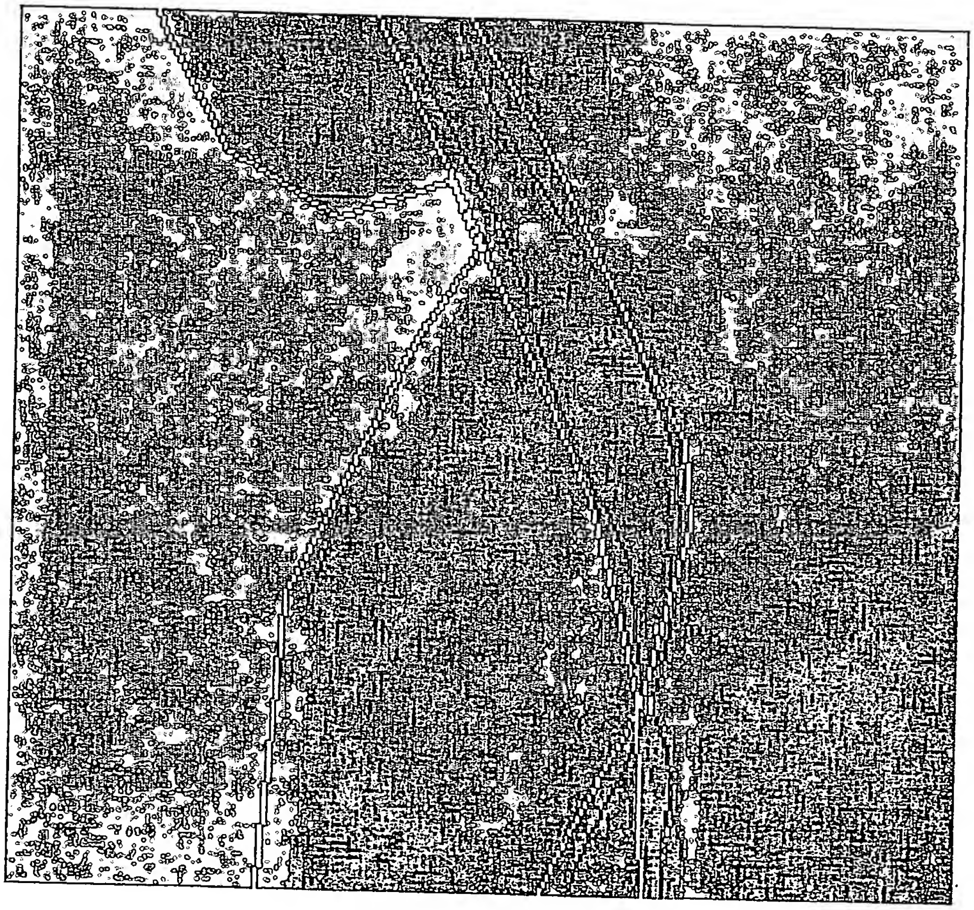


Figure 59

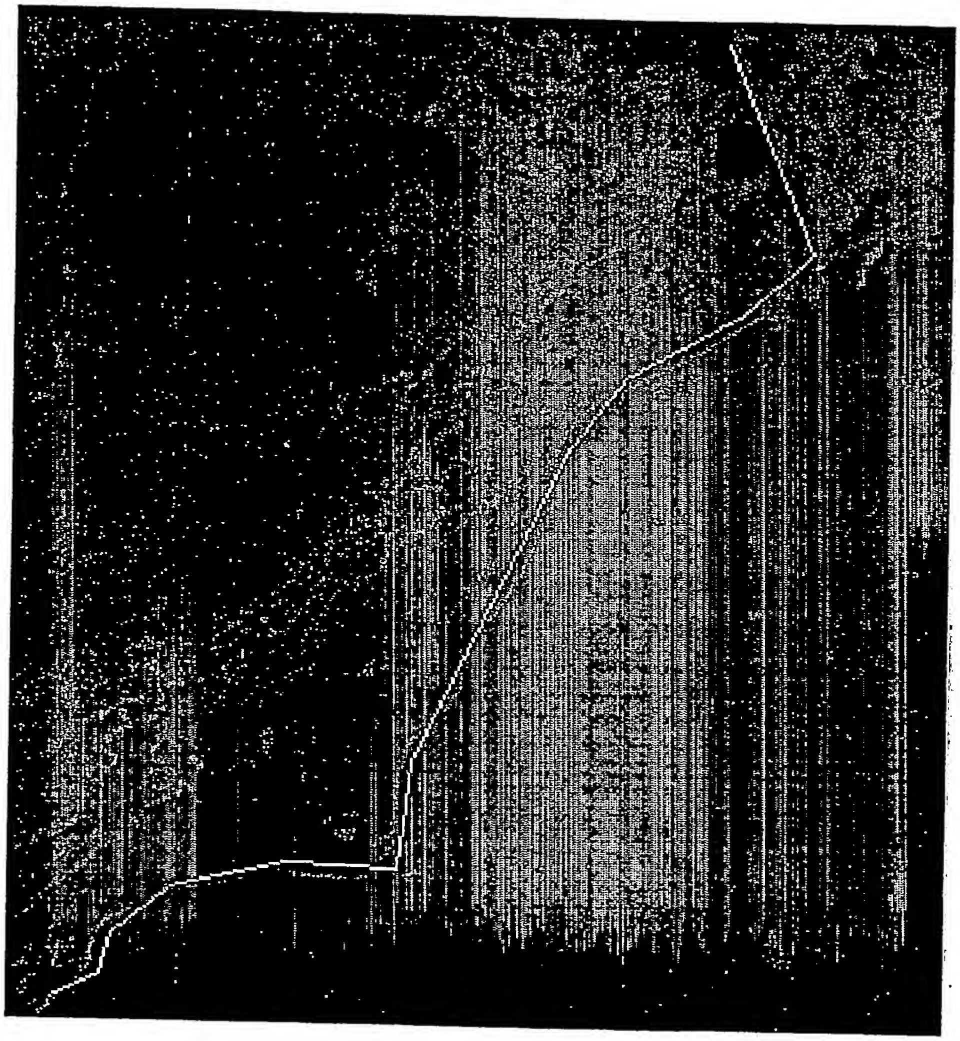


Figure 60

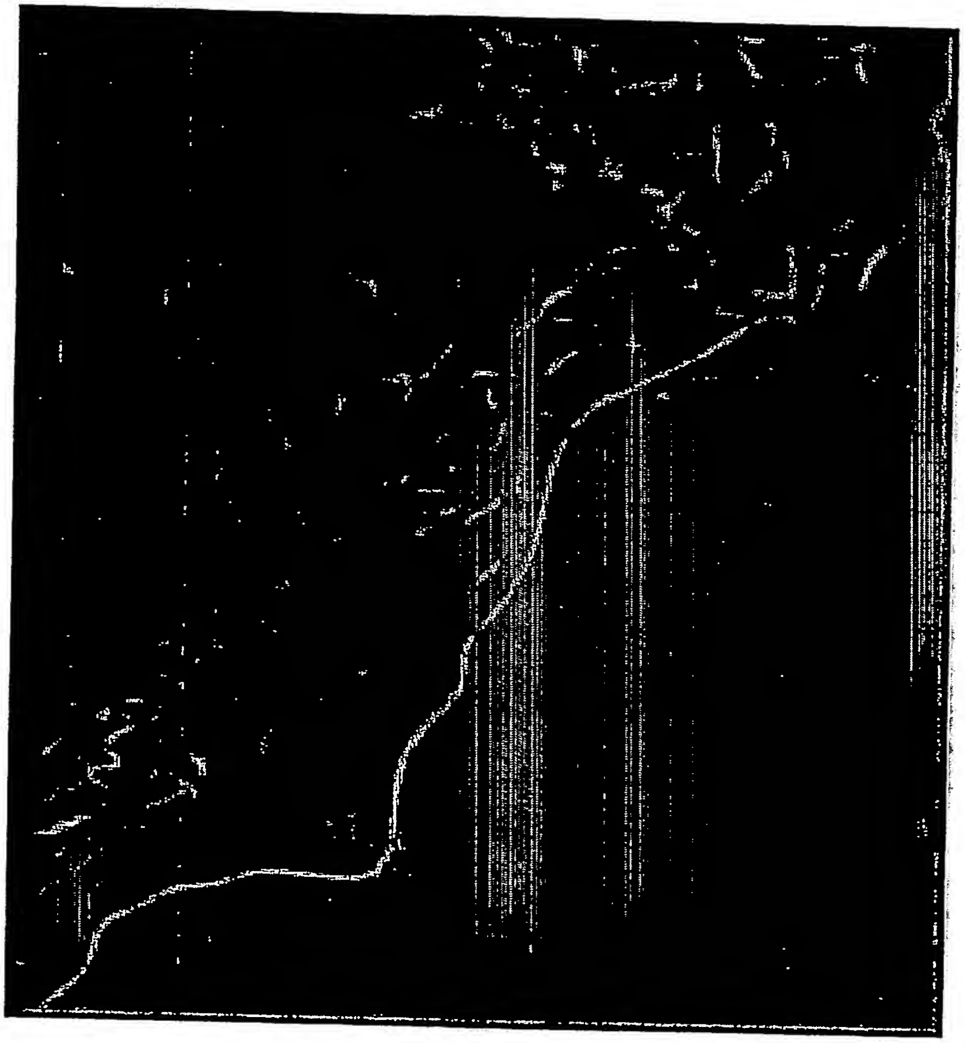


Figure 61

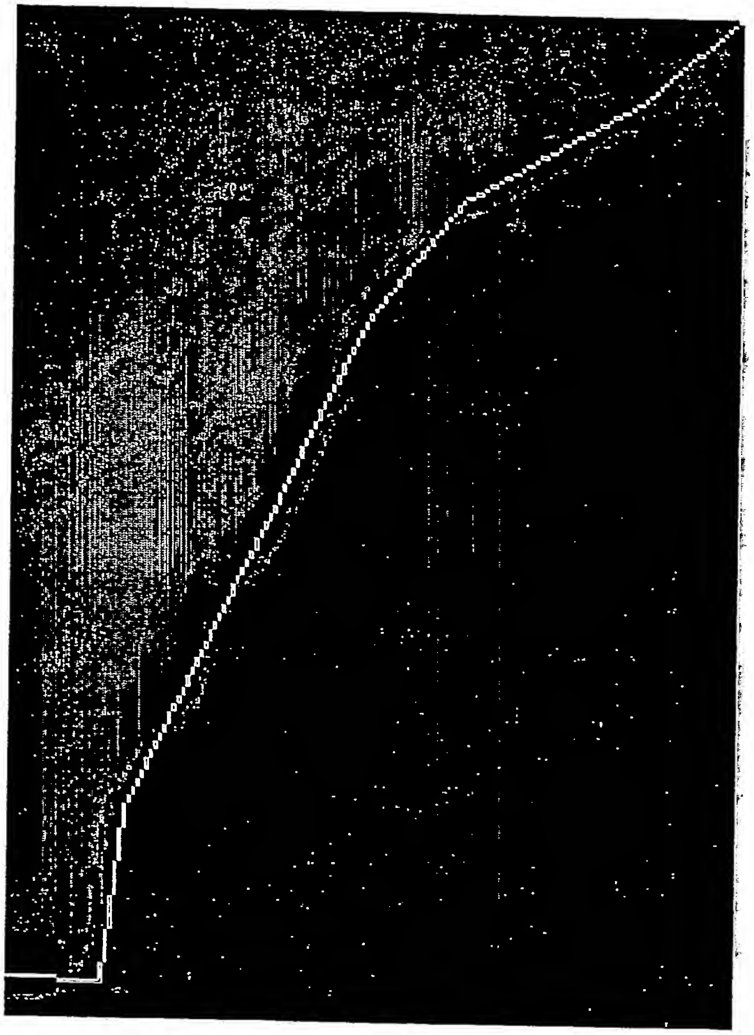


Figure 62

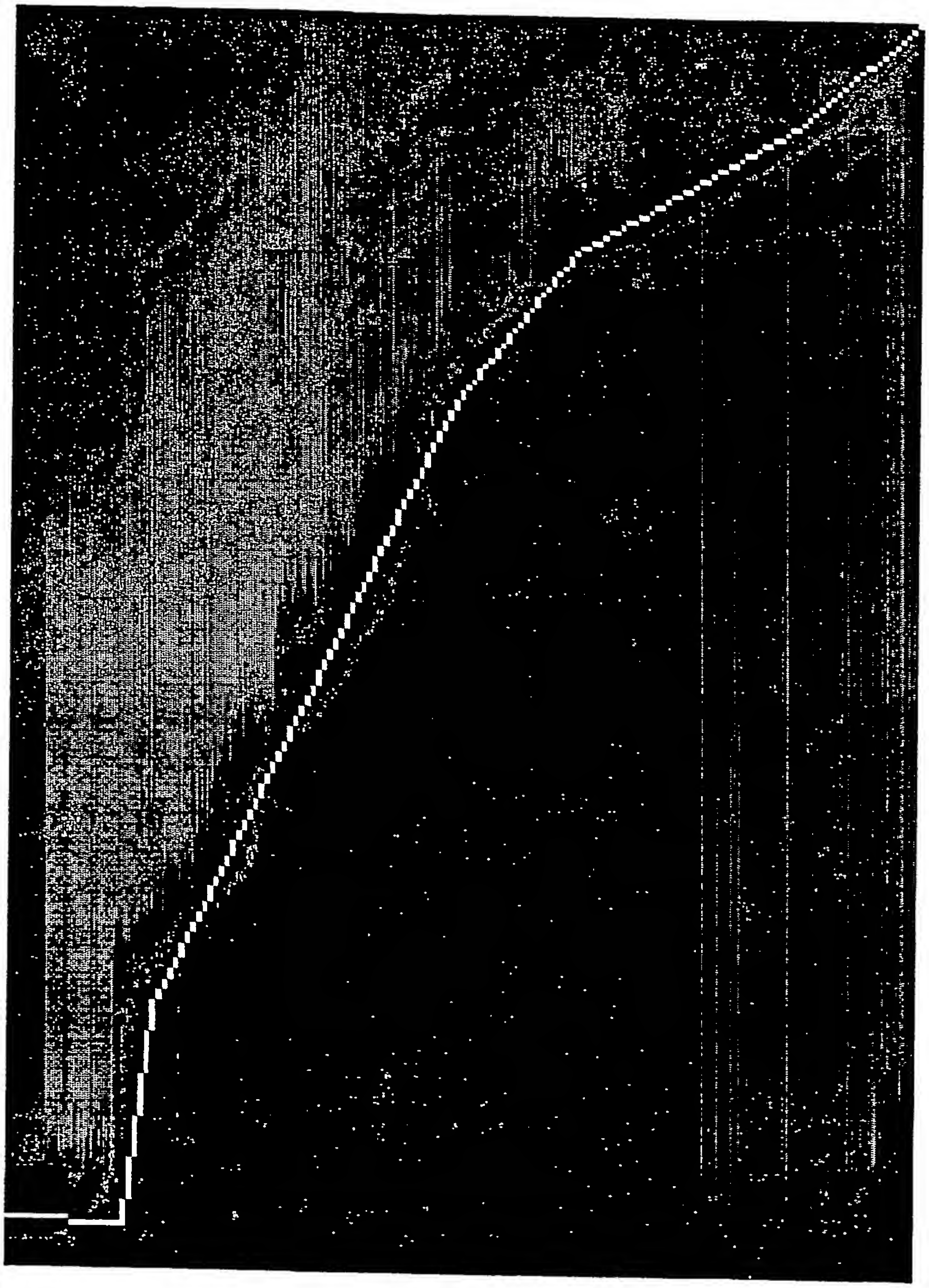


Figure 63

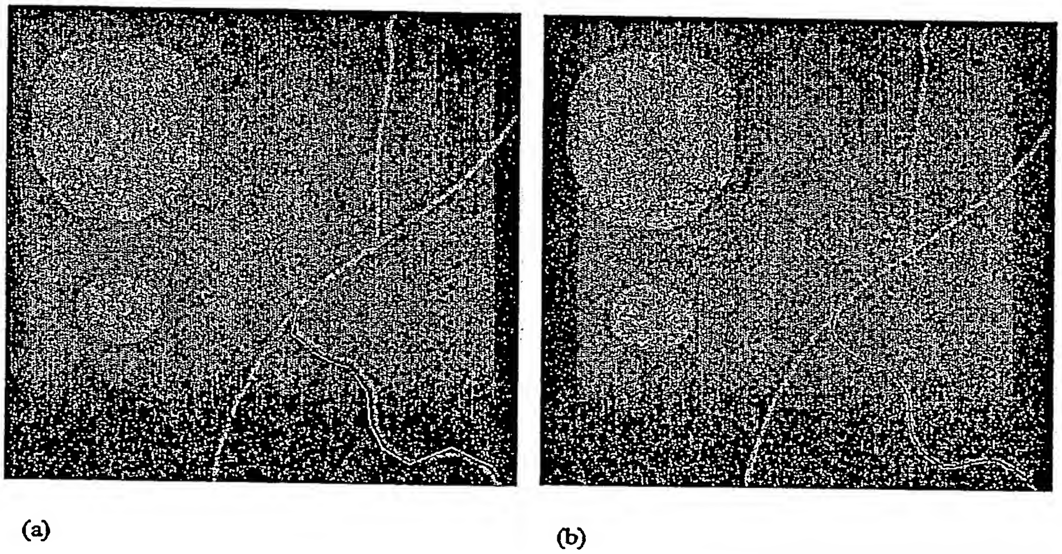


Figure 64

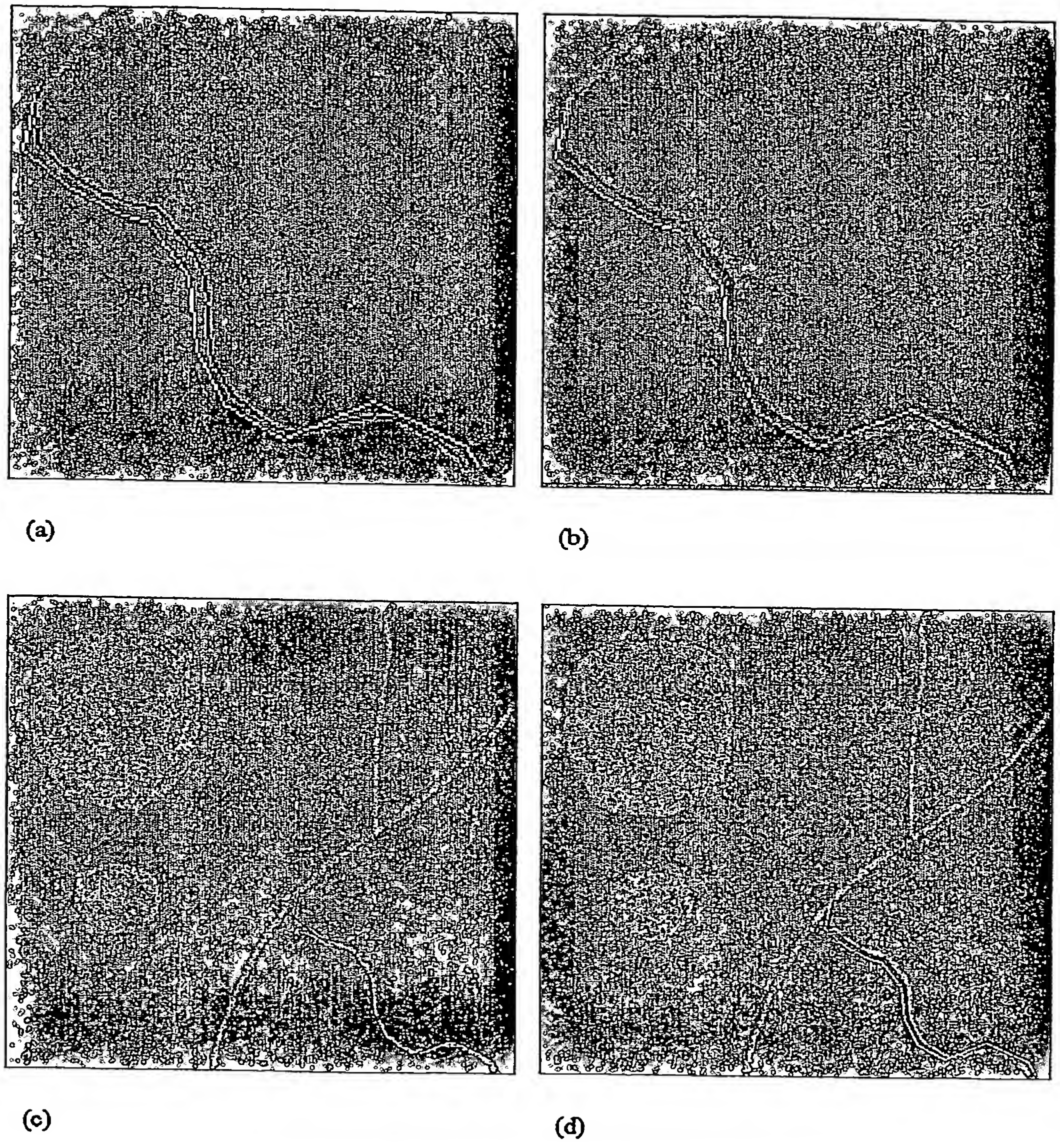


Figure 65